eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

Konputagailu Ingeniaritza eta Sistema Adimentsuak
Unibertsitate Masterra
Master Tesia

Testu eta irudien arteko antzekotasun semantikoa aztertzen

Egilea
Ander Salaberria Saizar

Zuzendaria
Eneko Agirre Bengoa

Zuzendarikideak
Oier Lopez de Lacalle Lekuona

informatika fakultatea
facultad de informática

2019

Laburpena

Lengoaia Naturalaren Prozesamendu eta Ikusmen Artifizialaren arloaren erdigunean modalitate anitzak, irudi eta testuak, aldi berean prozesatzea da helburu nagusietako bat.

Proiektu honetan modalitate anitzeko sistemen estimazioak modalitate bakarra erabiltzen dutenena baino hobeagoak diren aztertu da. Hori burutzeko testu eta irudien arteko antzekotasun semantikoak aztertu dira, *STS* eta *vSTS* atazen bidez. *STS* modalitate bakarreko ataza da, non bi esaldiren arteko antzekotasun semantikoak aztertzen diren. *IXA* taldeak berriki sortu duen *vSTS* atazan, aldiz, testu eta irudien errepresentazioak maneiatzen dira esaldien arteko antzekotasun semantikoak aztertzeko —hots, modalitate anitzeko ataza bat da—. Kasu honetan, esaldi bakoitza irudi batez lagunduta dator, esaldi hori irudiaren goiburukoa edo *captio*a delarik.

Ataza horiek ebazteko artearen egoeran dauden hainbat motatako neurona-sare sakon landu dira. Hauen artean modalitate bakarra erabiltzen duten *BERT*, *GloVe*, *GPT-2* eta *USE* sistemak aurkitzen dira, baita modalitate anitzeko errepresentazioak eraikitzen dituen *VSE++* sarea ere. Sistema hauek *STS* eta *vSTS* atazetara moldatu dira, hauen errendimendua kasu ez-gainbegiratu eta gainbegiratuetan aztertuz. Aipatutako sistemak *vSTS* datu multzoan probatu dira. Datu multzo hau proiektu honetatik kanpo sortu eta hedatu bada ere, bere garapena sakon aztertu dugu.

Gainera, modalitate anitzeko sistema berri bat sortu da, *DiscoGAN* arkitekturan oinarrituta dagoena, *DiscoGAN-M*^{3}. *DiscoGAN-M*^{3} sistemak modalitate anitzeko errepresentazioekin lan egiten ez badu ere, bi modalitateen arteko transformazioak ikasten ditu. Arkitektura berri honen logika, eraikuntza eta ikasketa prozesuak sakonki azaldu dira.

Gure lanak modalitate anitzeko adierazpenak erabiltzean emaitza hobeagoak lortzen direla erakusten du esperimentu ez-gainbegiratuetan. Hala ere, esperimentu gainbegiratuetan ez da horrela izan. Kasu horretan atentzio-mekanismoak erabiltzen dituen *BERT* bezalako *Transformer*rek emaitza onenak lortzen dituzte, *vSTS* atazan artearen egoera definituz. Egindako lanak modalitate anitzeko adierazpenen kontribuzioa erakusten badu ere, kasu gainbegiratuetan emaitzak hobetzeko aukera asko daudela uste da.

Eskertzeak

Lerro hauetan proiektua gauzatzen lagundu didaten guztioi nire eskerrik beroenak adierazi nahi dizkiet. Hauetatik aipamen berezi bat proiektuko zuzendariei —Eneko eta Oierri— eta proiektuan zehar lagundu didaten Gorka Azkune eta, batez ere, Aitor Soroari. Guztien ekarpen eta aholkuek proiektua behar bezala garatzea eta biribiltzea ahalbidetu dute.

Hau guztiagatik mila esker!

Gaien aurkibidea

Irudien aurkibidea

Taulen aurkibidea

1. KAPITULUA

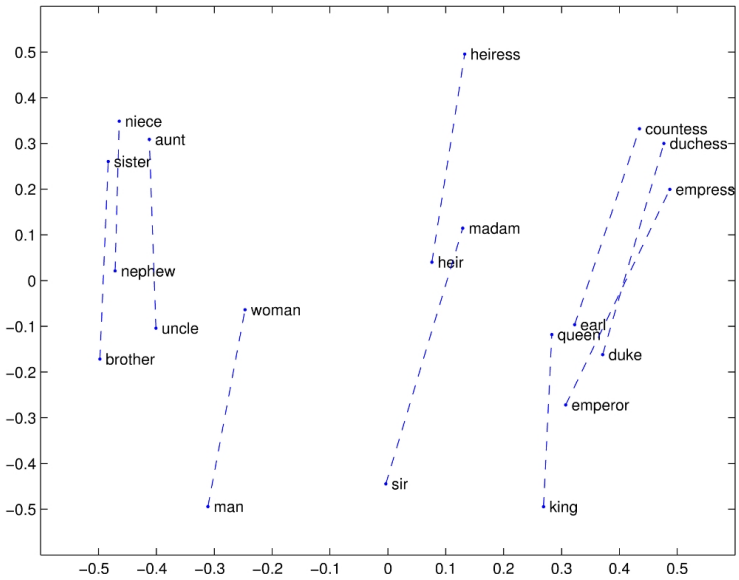
Sarrera

Hizkuntza naturalaren prozesamenduaren[1] arloan, *hitz-bektoreen* —hitzen errepresentazioak adierazten dituzten bektoreen— ikasketak arrakasta handia izan du. Errepresentazio hauek, *neurona-sareak* erabiliz, testu-corpus erraldoietatik ikasten dira. Neurona-sare hauek corpuseko hitz bakoitzari bere esanahiaren errepresentazio abstraktu bat esleitzen diote, ataza sinple batzuk ebazten saiatzen diren bitartean ikasten dituztelarik. *Hizkuntza-ereduen* ikasketa, adibidez, aipatutako ataza horri egokitzen zaio, non hitz sekuentzien gaineko probabilitate distribuzioak ikasteko prozesuan hitzen errepresentazio horiek lor daitezkeen.

Hitz-bektoreen artean hainbat operazio egin daitezkeela frogatu bada ere —1.1 irudiko hitzen arteko analogiak [1], adibidez—, hau ez da nahikoa hizkuntzaren prozesamendua hurrengo pausora eramateko. Hori dela eta, hitz-bektoreen ikasketak esaldien adierazpenak ikasteko antzeko metodoak sortzea motibatu duela esan daiteke [2]. *Esaldi-bektore* hauek ikasi eta prozesatzeak *hizkuntzaren ulermena* lortzeko oinarrizko pauso bat osatzen du; eta, pauso hori emateko, erdibideko beste ataza batzuk proposatu dira —*STS* [3] eta *NLI* [4], adibidez—, bi esaldiren arteko antzekotasun semantikoak aztertzen dituztenak.

Horien artean, *STS* edo *semantic textual similarity* [5] proiektu honetan landu den ataza da, esaldi pare bakoitzari 0 eta 5 arteko balio bat esleitzen diona. Balio hori handitzen den heinean, orduan eta antzekoagoak izango dira bi esaldi horiek semantikoki. Ataza hau makinen bidez automatizatzeak asko lagundu dezake pertsona eta makina arteko komu-

[1] Euskaraz letra etzanez idatzita dauden terminoen ingelesezko itzulpenak 6.1 eranskinean aurki daitezke. Itzulpen asko zuzenean egin badaitezke ere, irakurleak bere esanahia begiratu dezake zalantzarik izanez gero. Hitz hauen lehenengo agerpena bakarrik egongo da letra etzanez idatzita.



1.1 Irudia: Hitz pareen arteko erlazioak edo analogiak *GloVe* hitz-bektoreen espazioan —irudia [1] artikulutik hartuta—. Irudian, bektoreen dimentsionalitatea 300etik 2ra murriztu da, analogia hori grafikoki erakutsi ahal izateko.

nikazioan, baita —modu orokorrago batean— lehen aipatu den hizkuntzaren ulermen eta prozesamenduan ere.

Pertsonen arteko komunikazioan —hots, hizkuntza naturaletan—, informazioa ez da osoa, pertsonek aurretik duten ezagutzarekin osatu eta interpretatzen baitute. Makinek pertsonekin komunikatu ahal izateko ezinbestekoa da makinek ezagutza guzti hori bereganatzea; adibidez, ezagutza hori testu-multzo handietatik induzituz. Hala ere, testuetan munduari buruzko ezagutza nekez azaltzen da. Adibidez, zuhaitzak zer diren jakin gabe zuhaitzen deskribapenak irakurtzen badira, zaila da hauen forma, adarren itxura eta koloreak definitzea. Izan ere, deskripzio hauek gutxitan izaten dira nahikoak zuhaitz bat zehazki zer den jakiteko —aurrejakituririk gabe behintzat—, zuhaitz hori errealitatean zer den ikusi arte.

Berriki, gabezia horiek irudi eta bideo multzo handietatik ezagutza induzituz osatu nahi izan da, irudietan azaltzen den ezagutza bisuala erabiliz, eta testuen errepresentazioa mundu fisikoari lotuz irudien bitartez, *grounding* deritzon prozesuan. Testu eta irudien errepresentazioak espazio berera eramateko, hau da, *modalitate anitzeko* errepresentazio hauek lortzeko problema honi *language grounding problem* deritzo.

Hizkuntzaren prozesamenduan, aurreko arazoak eraginda, informazio bisualak erabiltzen dituzten ataza berriak definitu dira, hala nola: *irudi edo goiburuko berreskurapena*, *irudien goiburuko sorkuntza*, *bideo- eta testu-lerrokatzea* eta *VQA* —irudien gaineko galdera-erantzute ataza—.

Informazio bisual horiek kontuan hartuz, *STS* atazaren pareko ataza bat proposatu da, testu eta irudi bidezko antzekotasun semantikoa aztertzen duen ataza —edo *vSTS*—, non, bi esaldi aztertu beharrean, bi irudi eta esaldi pare konparatzen diren; hots, esaldi bakoitza irudi batez lagunduta dator, esaldi bakoitza dagokion irudiaren goiburukoa izanik. Beraz, *vSTS* eta *STS* atazen artean dagoen ezberdintasuna informazio bisual hori eskura izatea ala ez izatea da, hurrenez hurren.

Orain arte, *STS* ataza ebazteko eskura zeuden datu multzoak esaldi pareez osatutako instantziak zituzten bakarrik. Orain, aldiz, sortu berri den *vSTS* datu multzoa erabili daiteke *vSTS* atazaren abantaila teorikoak praktikan ebaluatu ahal izateko.

1.1 Helburuak

Proiektuaren helburu nagusia testu eta irudien arteko antzekotasun semantikoa neurtzen dituzten algoritmo eta datu multzoak aztertzea bada ere —testuak eta irudiak espazio berean errepresentatzeko alternatibak aztertuz ere bai—, proiektuko helburuak ondorengo puntuetan zatitu daitezke:

1. Hizkuntzaren prozesamenduko fundamentuak ikastea, ondorengo puntuak eginez:
   (a) "Deep Learning for Natural Language Processing" ikastaroan parte hartzea.
   (b) Hizkuntzaren prozesamenduko literatura irakurtzea.
2. Modalitate anitzeko sistemen estimazioak modalitate bakarra erabiltzen dutenena baino hobeagoak diren aztertzea, ondorengo puntuak burutuz:
   (a) *STS* eta *vSTS* atazetan artearen egoera definitzen duten tekniken analisia, bai modalitate bakarrekoak eta baita anitzekoak ere.
   (b) Aztertu diren teknikak eskura dauden datu multzoetan ebaluatzea, sortu berri den *vSTS* datu multzoa barne.
   (c) Neurona-sare sakon baten modalitate anitzeko arkitektura berri bat sortzea —arkitekturaren inplementazio, ikasketa-prozesu eta ebaluazioak barne—, antzekotasun semantikoen balioak emateko gai dena.

Puntu hauekin erlazionatuta, lan esparru ezberdinak erabili direla aipatu behar da. Proiektu honetan, *Python3* lengoaia erabili da, *Tensorflow* eta *PyTorch* liburutegien bitartez

neurona-sareak eraiki eta exekutatuz. Alde batetik, *Google Colab* plataforma erabili da, sortutako kodea exekutatzeko zerbitzariak eskaintzen dituena —12 orduko sesioetara mugatuta—. Plataforma hau denbora luzez exekutatu behar ez den kodea exekutatzeko erabili da, 12GB memoria duen NVIDIA Tesla K80 txartel grafikoa erabiltzailearen eskura uzten baitu eta. Gainera, kodea *Google Drive* biltegiratze-zerbitzuan gordetzea eta exekutatzea ahalbidetzen du, erabiltzearen bizitza erraztuz. Beste aldetik, exekuzio-denbora luzeak behar dituen kodea exekutatzeko, etxeko ordenagailua erabili da, 6GB-eko NVIDIA GeForce GTX 1060 txartel grafikoarekin. Etxeko ordenagailua ez da *Google Colab*eko zerbitzaria bezain azkarra; baina, etxeko "sesioak" denbora mugak ez dituenez, helburuetako azken puntuko arkitekturaren ikasketa-prozesua burutzeko aukerarik onena bihurtzen du.

1.2 Planifikazio eta jarraipena

Proiektua 2019ko otsailetik urte bereko irailera arte garatu da, nahiz eta tesia apirilera arte ia geldirik egon, ikasturteko klaseen lan-karga murriztu arte, hain zuzen ere. Lau atal nagusietan banatu daiteke proiektuaren garapena:

1. Hainbat argitalpen tekniko eta liburuen irakurketa, gaiari buruzko oinarrizko printzipio eta algoritmoak barneratzeko. Gainera, *Tensorflow* eta *PyTorch* liburutegiei buruzko hainbat dokumentazio eta adibide jarraitu dira, hauen erabilera zuzena ikasteko. Atal hau proiektu osoan zehar burutu da.

2. Aurreko puntuarekin batera, *Deep Learning for Natural Language Processing* mintegian parte hartu da otsailean zehar, hizkuntzaren prozesamenduaren eta neuronasare sakonen fundamentuak ikasteko intentzioarekin.

3. Aurretik aipatutako sistemekin, hainbat proba burutu dira *STS* eta *vSTS* atazetan, probak modu gainbegiratu eta ez-gainbegiratuetan burutuz. Atal hau, martxoa eta uztaila artean burutu da, nagusiki.

4. MultiDiscoGAN arkitekturaren diseinu eta inplementazioa, gehienbat maiatza eta ekaina artean burutua. Abuztuan, sistemari aldaketa batzuk egin zaizkio, bere errendimendua hobetzeko asmoarekin.

5. Dokumentazioaren idazketa proiektuan zehar burutzen joan da, gehienbat uztaila eta iraila artean garatu dena.

Bilerak astero egin dira zuzendari eta zuzendari-kideekin, hainbat salbuespen kontuan hartu gabe. Martxoan zehar, Masterreko irakasgaien lan karga zela eta, proiektuari geldiune bat eman zitzaion. Ekaina bukaeran eta uztailean zehar, doktoretzarako hainbat beka eskaera egin eta proiektua bigarren planoan utzi zen. Azkenik, abuztuan zehar, bilerak ez ziren ia burutu, gehienbat dokumentazioaren idazketa besterik ez baitzen geratzen.

1.3 Edukiak

Memoria honetan zehar, honako edukiak azalduko dira:

- Bigarren kapituluan, gaiari buruzko aurrekariak azalduko dira, bai erabili diren sistemak eta datu multzoak detailean azalduz.
- Hirugarren kapituluan, IXA taldean sortu berri den *vSTS* datu multzoaren sorkuntza eta ezaugarriak definituko dira.
- Laugarren kapituluan, *DiscoGAN-M*^{3} arkitektura berria nola jaio den azaldu eta bere egituraren zergatiak defendatuko dira.
- Bosgarren kapituluan, azaldutako sistema horiekin egin diren esperimentu, emaitza eta hausnarketak ikusiko dira.
- Seigarren kapituluan, emaitzetatik lortzen diren ondorioei buruz hitz egingo da, baita etorkizunerako gelditzen diren lanei buruz ere.
- Azkenik, lehenengo eranskinean, erabili den terminologiaren itzulpenak aurki daitezke.

2. KAPITULUA

Aurrekariak

Neurona-sare artifizialek ataza jakin batzuk ikasten dituzte, ataza horietako ebazpen eta adibideak kontsideratuz. Azkeneko hamarkadan indar handia hartu dute hainbat arloetan zehar, hizkuntzaren prozesamenduan barne; gehienbat, prozesadore eta txartel grafikoek izan dituzten garapenak eraginda. Horrek sistema berriak etengabe sortzea ekarri du, urtero hainbat atazetan artearen egoera definitzen dituzten sistema berriak publikatzen direlarik.

Hasiera batean, neurona-sareen helburu nagusia hauei emandako atazak gizakien burmuinak bezala ikastea bazen ere, denborak aurrera egin ahala, antzekotasun biologikoetatik urrundu da ataza espezifikoak ebazteko ahaleginean.

Hori dela eta, kapitulu honetan erabili diren neurona-sare —edo sistemak— eta datu multzoak komentatu dira. Sistema eta datu multzo hauek hainbat ataletan zatitu dira, hauek erabiltzen edo osatzen dituzten modalitateen arabera; hots, testua, irudia eta biak maneiatzen dituzten sistema eta datu multzoak atal ezberdinetan azaltzen dira. Sistema horiek barneratzen dituzten teknologiak ere gainetik azalduko dira, hainbat neurona-sare mota eta teknika aipatuz, besteak beste.

Ondorengo azpiataleko sistema eta datu multzo gehienak 5. kapituluan erabiltzen dira. Gainera, *STS* atazarekin zerikusia duten beste datu multzo batzuk gainetik azaldu dira.

2.1 Testuen errepresentazioak

Lehenengo azpiatal honetan, testua maneiatzen duten sistemak eta datu multzoak aipatzen dira. Sistema hauek esaldi-bektoreak sortzeko erabili dira, adibidez, esaldi-bektore hauek eraikitzeko esaldian agertzen diren hitzen hitz-bektoreak erabiltzen dituztenak.

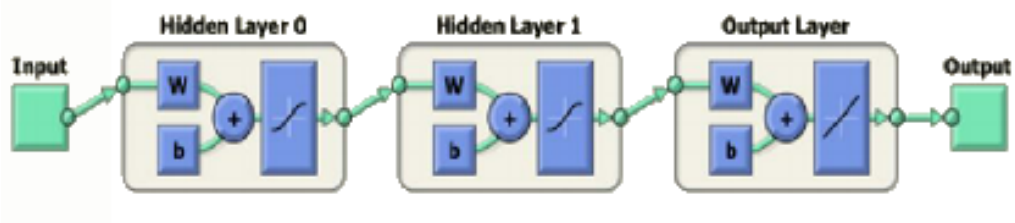
2.1.1 Oinarrizko sistemak

Proiektu honetan neurona-sare artifizialak erabili dira testu eta irudiak prozesatzeko orduan. Hitz gutxitan, neurona-sare artifizialak eredu matematikoak dira, gure burmuinaren barruan sortzen diren neuronen arteko erreakzio sinaptikoak imitatzen saiatzen direnak. Funtsean, sarrerako datuei funtzio matematiko bat aplikatzen die irteera datu batzuk lortzeko. Irteera datu horiek adierazgarriak izateko, sare hauek ikasketa-prozesu batetik pasa behar dira, bete nahi den atazaren arabera aldatzen dena.

Erabili diren sistemak bi ataletan bana daitezke: *neurona-sare errepikakor* edo *RNN*etan, eta *aurrerantz-elikatutako neurona-sareetan* oinarritutakoak —*Transformer*rak azkeneko hauetan sartzen direlarik—. Sistema bakoitza komentatu aurretik, erabilitako neurona-sare motak definituko dira. Hala ere, *RNN* eta *Transformer* hauek azaltzeko, aurrerantz-elikatutako neurona-sareak zer diren eta nola funtzionatzen duten gainetik errepasatzea ondo letorke, neurona-sare sinpleenetarikoak baitira.

Aurrerantz-elikatutako neurona-sareak (Feedforward Neural Network, FNN)

2.1 irudian aurrerantz-elikatutako neurona-sare hauen adibide baten egitura azaltzen da. Irudiko geruza hauek tipikoak dira neurona-sare hauetan, non geruza bakoitzak hiru eragiketa egiten dituen: matrizeen arteko biderketa, batuketa eta *aktibazio-funtzio* bat, hain zuzen ere.



2.1 Irudia: Aurrerantz-elikatutako neurona-sare baten arkitektura sinple bat, hiru geruzako *Multilayer-Perceptron* bati dagokiona.

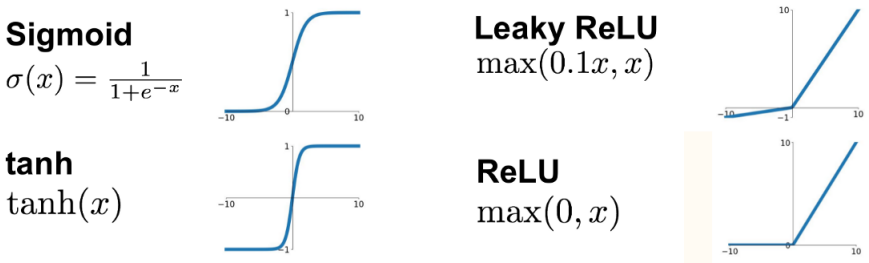
2.1 ekuazioan neurona-sare honetan ebazten diren hiru geruzen kalkuluak azaltzen dira —*geruza-dentsoak* bezala ezagutzen direnak—. Lehenengo geruzako sarrerako bektoreari $\mathbf{x}$ deituko zaio, eta azken geruzako irteerakoari, berriz, $\mathbf{y}$ —bi hauek matrizeak izan badaitezke ere, gauzak sinplifikatzeko bektoreak bezala definituko dira. G. geruzan, aurreko geruzako $\mathbf{h}^{(G-1)}$ irteera $\mathbf{W}^{(G)}$ matrizearekin biderkatu eta $\mathbf{b}^{(G)}$ bektorearekin batzen dira, $\mathbf{a}^{(G)} = \mathbf{h}^{(G-1)} \cdot \mathbf{W}^{(G)} + \mathbf{b}^{(G)}$ lortzeko.

$$\begin{aligned} \mathbf{y} &= \varphi(\mathbf{h}^{(1)} \cdot \mathbf{W}^{(2)} + \mathbf{b}^{(2)}) \\ \mathbf{h}^{(1)} &= \varphi(\mathbf{h}^{(0)} \cdot \mathbf{W}^{(1)} + \mathbf{b}^{(1)}) \\ \mathbf{h}^{(0)} &= \varphi(\mathbf{x} \cdot \mathbf{W}^{(0)} + \mathbf{b}^{(0)}) \end{aligned} \tag{2.1}$$

Matrize horien biderketa eta batuketak egiteko erabiltzen diren bektore eta matrizeen dimentsioak kontuan hartu behar dira —aljebra linealaren oinarrizko erregelak jarraituz—. Demagun G. geruzako sarrerako bektoreak a balioz osatuta dagoela, hots, $\mathbf{h}^{(g-1)}$ bektorea a elementuko bektorea dela. Geruzaren irteeran b elementuko bektore bat lortu nahi bada, $\mathbf{W}^{(G)}$-k $a \times b$ dimentsioko matrizea izan behar du nahitaez. Hau horrela izanik, $\mathbf{b}^{(G)}$ bektoreak b elementu izan beharko ditu batuketa egin ahal izateko.

Azkenik, $\mathbf{a}^{(G)}$-ri funtzio ez-lineal —edo aktibazio-funtzio— bat aplikatzen zaio, neurona-sarea ataza linealetara bakarrik ez mugatzeko.

Esan bezala, φ aktibazio-funtzioek neurona-sareei malgutasuna ematen die, funtzio ez-linealak ikasteko aukera ematen baitie. 2.2 irudian hainbat adibide ikus daitezke.



2.2 Irudia: Proiektu honetako esperimentuetan zehar erabili diren hainbat aktibazio-funtzio.

Neurona-sare askotako azken geruzan aktibazio-funtzio berezi bat erabiltzen da, lortzen diren balioak probabilitate bihurtzen dituelarik. *Softmax* deitzen den funtzio hau oso baliagarria da —sailkatzaileetan batez ere—, klase bakoitzak duen probabilitatea lortzeko metodo eraginkorra baita —2.2 ekuazioa—.

$$\sigma(\mathbf{y})_j = \frac{e^{\mathbf{y}_j}}{\sum_{k=1}^{K} e^{\mathbf{y}_k}} \tag{2.2}$$

Sarrerako **x** bektoretik irteerako **y** bektorea lortzeari, hots, neurona-sareak sarrerako datu batzuen estimazioei *aurreranzko propagazioa* deritzo. Baina ataza hau ondo egiteko, sarea ikasketa-prozesu batetik pasa behar da.

Sareak ikasten

Neurona-sareen ikasketa hasi aurretik, **W** eta **b** guztien balioak ausaz definitzen dira. Ondoren, aurreranzko propagazio bakoitzeko *atzeranzko propagazio* bat egiten da ikasketa fasean.

Atzeranzko propagazioa burutzeko galera-funtzioak erabiltzen dira, neurona-sarearen errendimendua neurtzen duena —optimizazio problema baten *helburu-funtzio* gisa—. Neurona-sareak **t** benetako emaitzei emandako estimazio on edo txarren arabera, galera-funtzioaren balioa handiagoa ala txikiagoa bihurtuko da, hasieran definitutako ikasketa-ataza minimizazio problema batera murriztuz. Galera-funtzio baten adibidea jartzeagatik, estimazioekin lortzen den batez-besteko errore-koadratikoa erabili daiteke erregresio problemetan —2.3 ekuazioa—.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{y}_i - \mathbf{t}_i)^2 \tag{2.3}$$

Atzeranzko propagazioan $\mathbf{W}^{(G)}$ eta $\mathbf{b}^{(G)}$ pisuen balioak emaitza hobeagoak lortzeko eraldatzen dira. Azken propagazio hau *gradiente-jaitsiera* bidez egiten da —funtzio baten minimoa kalkulatzen duen lehen ordenako optimizazio metodoa—; modu eraginkor batean exekutatzen dena *katearen erregela* erabiliz. Funtsean, gradiente-jaitsieraren metodoa *galera-funtzioaren* deribatuak sarearen pisu bakoitzarekiko kalkulatzean datza —2.4 ekuazioa—, deribatu horiek pisuak aldatzeko erabiliz.

$$\begin{aligned}
\Delta\mathbf{W}_{ij}^{(G)} &= -\eta\frac{\partial L}{\partial \mathbf{W}_{ij}^{(G)}} = -\eta\frac{\partial L}{\partial \mathbf{h}_j^{(G)}}\frac{\partial \mathbf{h}_j^{(G)}}{\partial \mathbf{a}_j^{(G)}}\frac{\partial \mathbf{a}_j^{(G)}}{\partial \mathbf{W}_{ij}^{(G)}} = -\eta\mathbf{h}_i^{(G-1)}\delta_j^{(G)} \\
\Delta\mathbf{b}_j^{(G)} &= -\eta\frac{\partial L}{\partial \mathbf{b}_j^{(G)}} = -\eta\frac{\partial L}{\partial \mathbf{h}_j^{(G)}}\frac{\partial \mathbf{h}_j^{(G)}}{\partial \mathbf{a}_j^{(G)}}\frac{\partial \mathbf{a}_j^{(G)}}{\partial \mathbf{b}_j^{(G)}} = -\eta\delta_j^{(G)}
\end{aligned} \tag{2.4}$$

Pisuen aldaketak galera-funtzioaren balioari eragiten dion aldaketa kontrolatzeko *ikasketa-tasa* edo η erabiltzen da. Ekuazioetan erabilitako nomenklaturaren adibide bezala, $\mathbf{W}_{ij}^{(G)}$ aldagaiak G geruzan $\mathbf{W}$ matrizeak i. lerroan eta j. zutabean duen balioa adierazten du. Ikasketa-prozesua minimizazio problema bat denez, 2.4 ekuazioan zeinu negatiboa jartzen da —maximizazio problema balitz, ez litzateke zeinu hori agertuko—.

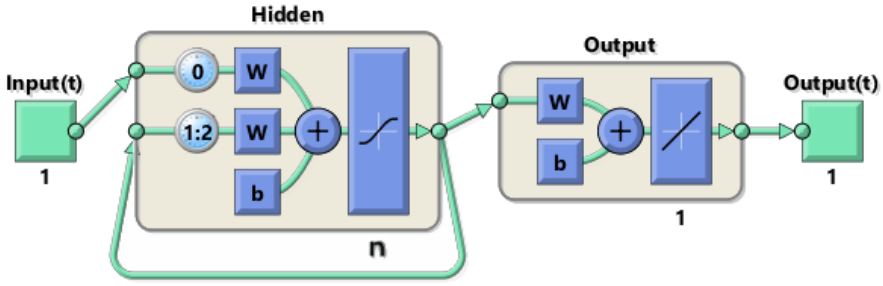
$$\delta_j^{(G)} = \frac{\partial L}{\partial \mathbf{h}_j^{(G)}} \frac{\partial \mathbf{h}_j^{(G)}}{\partial \mathbf{a}_j^{(G)}} = \begin{cases} 2(\mathbf{h}_j^{(G)} - \mathbf{t}_j^{(G)}) \cdot \varphi'(\mathbf{a}_j^{(G)}) & \text{G azken geruza bada,} \\ \left(\sum_{k \in \mathbf{h}^{(G+1)}} \mathbf{W}_{jk}^{(G)} \cdot \delta_k^{(G+1)}\right) \cdot \varphi'(\mathbf{a}_j^{(G)}) & \text{bestela.} \end{cases} \tag{2.5}$$

2.5 ekuazioan definitu den δ_j-ren kalkuluan galera-funtzioa errore koadratikoa dela kontuan hartu da —2.3 ekuazioa—. Erregresio problema baterako galera-funtzioa aukeratzean *batez-besteko errore koadratikoa* oso erabilia da; baina, kontuan izan behar da galera-funtzioen aukeraketa ebatzi nahi den atazaren menpekoa dela.

Sare hauek oso onak dira **x** sarrera eta **y** irteera pareen arteko erlazioak ikasten, baina datuen ordenak garrantzia duenean —hizkuntza-eredu bat ikastean, adibidez—, neurona-sare errepikakorrak erabiltzea komeniko litzateke. Izan ere, aurrerantz-elikatutako neurona-sare askok ez dute datuen ordena kontuan hartzen, hau da, ez du aurreko datuen informaziorik gordetzen.

Neurona-sare errepikakorrak (Recurrent Neural Network, FNN)

Irakurleak dokumentu hau irakurtzen duen heinean, hitz bakoitzaren esanahia ulertzeko aurretik irakurritakoa kontuan hartzen du. Beste modu batera esanda, gure pentsamenduak iraunkorrak dira denboran zehar, bizi ditugun gertaerak memorizatzen ditugularik.



2.3 Irudia: *RNN* baten egitura, *RNN*ko *gelaxkaren* irteera-bektore bakoitza geruza bateko aurrerantz-elikatutako neurona-sare batetik pasatzen delarik.

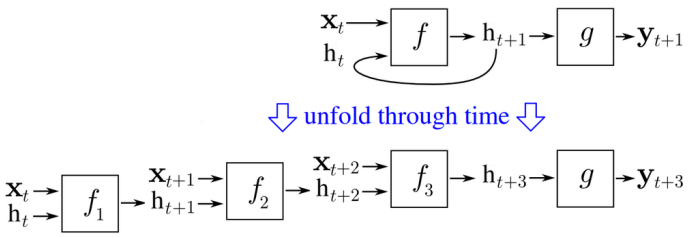
Neurona-sare errepikakorrek —*RNN*— neurona-sareen azpimultzo bat osatzen dute, informazioaren iraunkortasuna ahalbidetzen duen begiztak dituztelarik —2.3 irudia—. Hau horrela izanik, sare hauen nodoen konexioek grafo zuzendu bat eraikitzen dutela esan daiteke denboraren ardatzean zehar; hots, sare honek itzultzen dituen emaitzak aurretik aztertu dituen datuen menpekoak direla. Honek n luzerako hitz-bektore sekuentzietatik esaldibektoreak eraikitzeko egokiak bihurtzen ditu, hitz-bektore horiek aurrerantz-elikatutako neurona-sareak —edo beste neurona-sare mota— erabiliz kalkulatu daitezkeelarik.

Esan bezala, *RNN*ak aurretik ikusitako datuak memorizatzeko gaitasuna dute, sekuentzia bat jarraitzen duten datuak prozesatu ahal izateko. Memorizatze hori aurretik datozen datuen irteera-bektorea hurrengo instantziarekin batera prozesatuz lortzen da, ondoren azaltzen den bezala. Aurrerantz-elikatutako neurona-sareak bezala, sare hauek f funtzio bat aplikatzen dute instantzia bakoitzean; matrizeen arteko biderketak, batuketak eta aktibazio-funtzio baten aplikazioak osatzen dituena.

RNN hauentzako nomenklatura zerbait aldatzen da, hainbat geruza erabili beharrean geruza bera exekutatuko delako t-ren balio ezberdinetarako. Hori dela eta, $\mathbf{x}^{(t)}$ eta $\mathbf{h}^{(t)}$ adierazpenek t hitzaren sarrera eta irteera-bektoreak adieraziko dituzte, hurrenez hurren. $\mathbf{W_r}$, berriz, $\mathbf{x}^{(t)}$ eta $\mathbf{h}^{(t-1)}$ bektoreen konkatenazioarekin biderkatuko diren pisuen matrizea izango da; eta, azkenik, $\mathbf{b_r}$ bektorea aurreko biderkaduraren emaitzekin batera batzen da, $\mathbf{a}^{(t)}$ lortzeko —2.6 ekuazioa—.

$$\mathbf{h}^{(t)} = \varphi(\mathbf{a}^{(t)}) \longrightarrow \mathbf{a}^{(t)} = \mathbf{W_r} \cdot \left[\mathbf{h}^{(t-1)}, \mathbf{x}^{(t)}\right] + \mathbf{b_r} \tag{2.6}$$

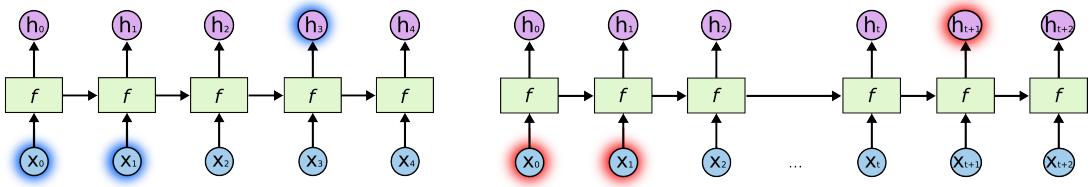
2.6 ekuazioaren kasu nabaria edo $t = 0$ kasua kodetzeko $\mathbf{h}^{(-1)}$ zeroz osatutako bektore bezala hasieratzen da. Sare hauetan $\mathbf{x}^{(t)}$ eta $\mathbf{h}^{(t)}$ bektoreek a eta b elementu dituztela kontuan hartzen bada, hurrenez hurren; $\mathbf{W_r}$ matrizeak $(a+b) \times b$-ko dimentsionalitatea izango du; eta, azkenik, $\mathbf{b_r}$ bektoreak b elementu izango ditu.



2.4 Irudia: t instantziako sarearen memoria eta sarrerako datuek duten erlazioa azaltzen da, aurreko instantzien datuekiko. Irudi honetan eta ondorengoetan, $\mathbf{x}_t$ adierazpena dokumentuan zehar erabiltzen den $\mathbf{x}^{(t)}$-ren parekoa da.

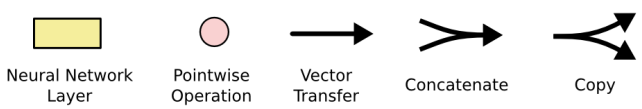
RNN hauetan atzeranzko propagazioa aplikatzerakoan, behin eta berriro errepikatzen den *RNN gelaxka hedatu* behar da —2.4 irudian azaltzen den f funtzioa kalkulatzen duen gelaxka hedatu behar delarik—. Behin hedapen hori eginda, atzeranzko propagazioa aurrerantz-elikatutako neurona-sare bat balitz bezala egin behar da —2.4 ekuazioari hainbat aldaketa eginik, egiten diren eragiketak ez baitira guztiz berdinak—. Atzeranzko propagazio honi *BackPropagation Through Time* edo *BPTT* deritzo. Gainera, neurona-sare guztietan bezala gradiente-jaitsiera metodoak erabiliko direnez, kontuz ibili behar da kalkulatzen diren deribatu partzialak oso handiak ala oso txikiak ez bihurtzearekin —problema hauei ingelesez *exploding-gradient* eta *vanishing-gradient* deitzen zaie, hurrenez hurren—.

Azaldu den *RNN* hau 1980. hamarkadan definitu zen oinarrizko neurona-sare errepikakorra da. Sare honek hizkuntza-eredu bat ikasteko zailtasunak izango ditu esaldietan aurki daitezkeen hitzen arteko mendekotasunen arabera. Kontsidera dezagun hizkuntza-eredu bat, esaldi bati falta zaion h hitza iragarri behar duena aurreko hitzak aurreprozesatuz. h aurreko hitzen menpekoa da, baina h iragartzeko behar diren hitz gakoak oso urruti egon daitezke iragarri nahi den hitzetik. Oinarrizko *RNN* sareek azkeneko iterazioetan prozesatu diren datuak dituzte gogoan gehienbat, hitzen arteko distantzia handiko mendekotasunak ikasi ezinik. Sare hauek epe laburreko memoria dutela esaten da —2.5 irudia—.



2.5 Irudia: Lehenengo irudian urdinez koloreztatua agertzen diren sarrerako eta irteerako datuen artean menpekotasuna antzematea errazagoa da bigarren irudiaren kasuan baino, menpekotasun hori azaltzen den datuak urrutiago baitaude. Ikus daitekeenez, sekuentziako aurreko datuen eragina ahazten edo diluitzen doaz hurrengo datuak prozesatzen diren heinean.

Hori dela eta, instantzia bakoitzean 2.6 ekuazioak aplikatu beharrean, hots, azaldu den 2.3 irudiko oinarrizko gelaxka erabili beharrean, beste gelaxka mota batzuk erabili daitezke, sare errepikakor hauen memoria hobetzeko ahaleginean. Gelaxka hauek azaltzerako orduan hainbat eskema erabiliko direnez, eskemetako eragiketak 2.6 irudian agertzen dira.

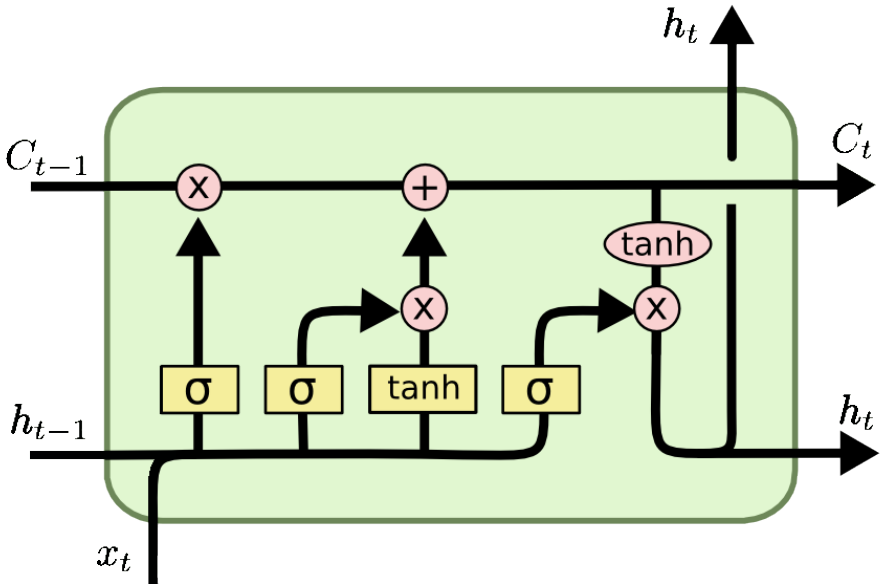


2.6 Irudia: 2.7 eta 2.8 irudietako eskemen notazioa. Notazio hau erabiltzen dituzten irudiak [6] artikuluan azaltzen dira.

Long-Short Term Memory

Sare errepikakor mota hauek *Long-Short Term Memory* edo *LSTM* deritze [7], epe labur eta luzeko menpekotasunak ikasteko gai dira eta. Izatez, epe luzeko menpekotasunen arazoa saihesteko diseinatu ziren.

Oinarrizko *RNN* gelaxketan bezala, f funtzio bat begizta baten bidez errepikatzen da, iterazio bakoitzean $\mathbf{x}^{(t)}$ sarrerako datu berri bat eta aurreko iterazioko $\mathbf{h}^{(t-1)}$ irteerako datuarekin kalkuluak eginik. Baina, oro har, aipatu den arazoa ekiditeko asmoarekin f konplexuagoa bihurtzen da —2.7 irudia—.



2.7 Irudia: *LSTM* gelaxkak burutzen duen f funtzioaren eskema.

LSTM-en gakoa $\mathbf{C}^{(t)}$ *egoera-gelaxka* edo *cell-state* da, epe luzeko memoriaz arduratzen dena. Iterazio bakoitzean bektore honen balioa aldatu egiten da, ikasi edo ahaztu nahi den informazioaren arabera. Egoera-gelaxka hau $\mathbf{h}^{(t)}$ definitzeko erabiltzen da, lehen aipatu diren balioekin batera.

Ikasketa ala ahazte-prozesu hauek ateak —*gate*— deituriko egituren bidez burutzen dira. Hiru ate ezberdin erabiltzen ditu *LSTM* gelaxkak iterazio bakoitzean: ahazte-atea edo *forget-gate*, sarrerako atea edo *input-gate* eta irteerako atea edo *output-gate*.

- Ahazte-atea: t instantzian $\mathbf{C}^{(t-1)}$ egoera-gelaxkak zer ahaztuko duen aukeratzen du. Ahazte hori $\mathbf{h}^{(t-1)}$ eta $\mathbf{x}^{(t)}$ bektoreen araberakoa da. Funtsean, $\mathbf{C}^{(t-1)}$ bektoreko balio bakoitza 0 eta 1 arteko zenbaki batekin biderkatuko da —0z bidertzen bada, balio hori ahaztu egingo da; 1ekin bidertzen bada, berriz, balio hori mantendu

egingo da—. 0 eta 1 arteko balio horiek $\mathbf{f}^{(t)}$-ren kalkuluarekin lortzen dira —2.7 ekuazioa—.

$$\mathbf{f}^{(t)} = \sigma(\mathbf{W_f} \cdot [\mathbf{h}^{(t-1)}, \mathbf{x}^{(t)}] + \mathbf{b_f}) \tag{2.7}$$

- Sarrerako atea: Ate honek $\mathbf{h}^{(t-1)}$ eta $\mathbf{x}^{(t)}$ bektoreetatik zer ikasiko den kalkulatzen du. Ikaste-prozesu honek bi atal dauzka. Alde batetik, $\tilde{\mathbf{C}}^{t-1}$ bektoretik zein balio mantenduko diren aukeratzen duen geruza du, $\mathbf{f}^{(t)}$-ren antzera kalkulatzen dena eta $\tilde{\mathbf{C}}^{(t)}$ bezala izendatzen dena. Beste aldetik, $tanh()$ aktibazio-funtzioa erabiltzen duen geruza-dentso bat dauka, $\tilde{\mathbf{C}}^{t}$ bektorea uneko egoera-gelaxkan eguneratu nahi diren balioak kalkulatzen direlarik. Bi geruza hauek 2.8a eta 2.8b ekuazioetan definitzen dira.

$$\mathbf{i}^{(t)} = \sigma(\mathbf{W_i} \cdot [\mathbf{h}^{(t-1)}, \mathbf{x}^{(t)}] + \mathbf{b_i}) \tag{2.8a}$$
$$\tilde{\mathbf{C}}^{(t)} = \tanh(\mathbf{W_C} \cdot [\mathbf{h}^{(t-1)}, \mathbf{x}^{(t)}] + \mathbf{b_C}) \tag{2.8b}$$

Behin bi funtzio hauek burutzen direnean, $\mathbf{C}^{(t-1)}$ egoera-gelaxka eguneratu daiteke 2.6 ekuazioan agertzen den bezala. Ekuazio hauetan $*$ ikurrak matrizeen elementuen arteko biderkadura adierazten du.

$$\mathbf{C}^{(t)} = \mathbf{f}^{(t)} * \mathbf{C}^{(t-1)} + \mathbf{i}^{(t)} * \tilde{\mathbf{C}}^{(t)} \tag{2.9}$$

- Irteerako atea: Azkenik, kalkulatu berri den $\mathbf{C}^{(t)}$ egoera-gelaxka erabiliko da $\mathbf{h}^{(t)}$ irteera-bektorea kalkulatu ahal izateko. Hasteko, $\sigma()$ aktibazio-funtzioa duen geruza-dentso bat erabiliko da —$\mathbf{f}^{(t)}$ eta $\mathbf{i}^{(t)}$-ren egitura bera duena eta $\mathbf{o}^{(t)}$ deituko dena—, irteerako bektorean uneko egoera-gelaxkaren zein balio azalduko diren jakiteko. Ondoren, egoera-gelaxkaren balioei $tanh()$ aplikatuko zaie, irteera-bektoreko balio guztiak (-1, 1) tartera mugatzeko. Azkeneko bi eragiketa hauek 2.10a eta 2.10b ekuazioetan azaltzen dira.

$$\mathbf{o}^{(t)} = \sigma(\mathbf{W_o} \cdot [\mathbf{h}^{(t-1)}, \mathbf{x}^{(t)}] + \mathbf{b_o}) \tag{2.10a}$$
$$\mathbf{h}^{(t)} = \mathbf{o}^{(t)} \cdot \tanh(\mathbf{C}^{(t)}) \tag{2.10b}$$

Esan bezala, epe luzeko menpekotasunak ikasteko gaitasuna du neurona-sare errepikakor mota honek, baina ikasi behar den pisu kopurua oinarrizko *RNN*-ena baino askoz handiagoa da.

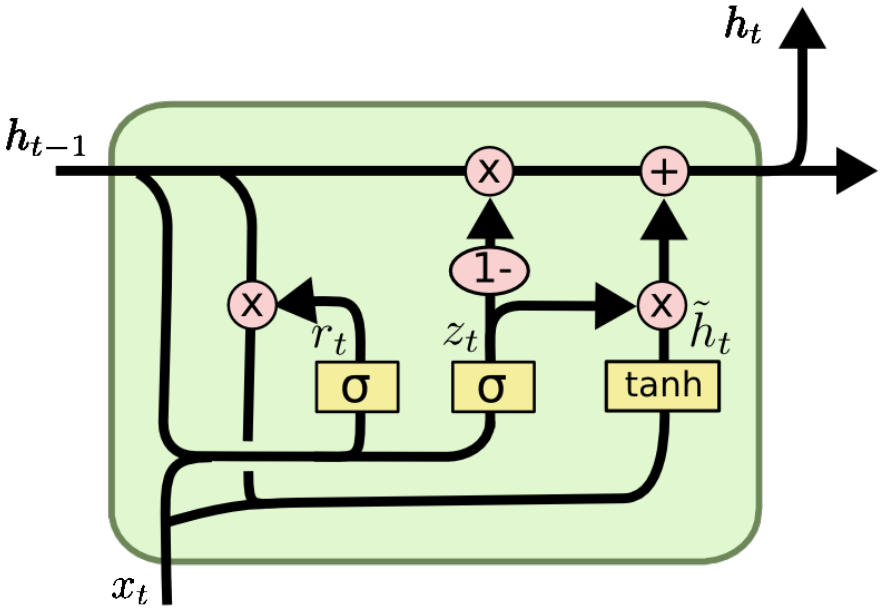
Ikasi beharreko pisuen artean dimentsio bereko lau **W** matrize eta beste lau **b** bektore daudenez, oinarrizko *RNN* gelaxkaren pisu kopurua lau aldiz handitzen du *LSTM*-ak, ikasketa-prozesuari kostu altuagoa emanik.

LSTM-ei buruz bukatzeko, urteetan zehar sare hauei buruzko lan askotan aldaketa txikiak aplikatu zaizkie —[8], adibidez—; baina, funtsean, egitura bera dute. Konparazio asko egin dira *LSTM* ezberdinen artean [9] [10], baina hauen arteko errendimendua atazaren menpekoa izan ohi da, eta aldaketak ez dira ia nabaritzen.

Gated Recurrent Unit

Epe luzeko menpekotasunak mantentzen dituzten *RNN*-en artean *Gated Recurrent Unit* edo *GRU* neurona-sare errepikakorrak antzeko errendimendua du, ikasi behar dituen pisu kopurua txikituz.

Hasteko, $\mathbf{C}^{(t)}$ eta $\mathbf{h}^{(t)}$ bektoreak konbinatzen ditu. Bere egitura 2.8 irudian ikus daiteke eta bere erabilera asko handitu da azkeneko urteotan, gelaxka mota honen sinpletasunak eraginda.



2.8 Irudia: *GRU* gelaxkak burutzen duen funtzioaren eskema.

Gelaxka honen ateak edo faseak honela deitzen dira: eguneraketa-atea edo *update-gate*, berrezartze-atea edo *reset-gate* eta irteerako bektorearen kalkulua. Bakoitzak burutzen duen ataza ondoren definitzen da.

- Eguneraketa-atea: Eguneraketa-ateak aurretik jaso duen informazioa hurrengo iteraziora zenbateraino pasatzea behar duen zehazten du. Aurrerantz-elikatutako geruza sinple bat da, σ aktibazio-funtzioa erabiltzen duena. Atea honek itzultzen duen bektorea $\mathbf{z}^{(t)}$ bezala izendatzen da —2.11 ekuazioa—.

$$\mathbf{z}^{(t)} = \sigma(\mathbf{W_z} \cdot \left[\mathbf{h}^{(t-1)}, \mathbf{x}^{(t)}\right] + \mathbf{b_z}) \tag{2.11}$$

- Berrezartze-atea: Sareak ate honen aurretik jaso duen zein informazio ahaztu behar duen definitzen du. Eguneraketa-atearen egitura bera badu ere —2.12 ekuazioa—, bakoitzaren pisuak helburu desberdinetarako definitzen dira sarearen ikasketa-fasean. Ate honek itzultzen duen bektorea $\mathbf{r}^{(t)}$.

$$\mathbf{r}^{(t)} = \sigma(\mathbf{W_r} \cdot \left[\mathbf{h}^{(t-1)}, \mathbf{x}^{(t)}\right] + \mathbf{b_r}) \tag{2.12}$$

- Irteerako bektorearen kalkulua: Azkeneko fase honetan $\tilde{\mathbf{h}}^{(t)}$ aldiuneko irteera-bektorea kalkulatzen da berrezartze-atea erabiliz —2.13a ekuazioa—; eta, ondoren, irteerako bektorea kalkulatzen da eguneraketa-atearen bidez —2.13b ekuazioa—.

$$\tilde{\mathbf{h}}^{(t)} = \sigma(\mathbf{W_h} \cdot \left[\mathbf{r}^{(t)} * \mathbf{h}^{(t-1)}, \mathbf{x}^{(t)}\right] + \mathbf{b_h}) \tag{2.13a}$$

$$\mathbf{h}^{(t)} = (1 - \mathbf{z}^{(t)}) * \mathbf{h}^{(t-1)} + \mathbf{z}^{(t)} * \tilde{h}^{(t)} \tag{2.13b}$$

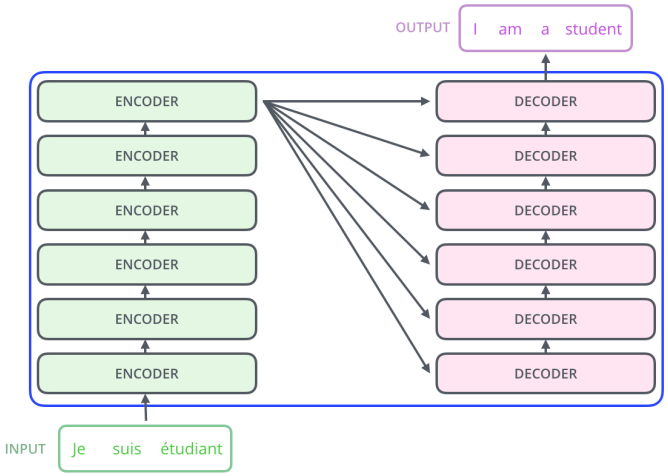
Guztira, *LSTM* gelaxkarekin konparatuz, *GRU* gelaxkak erabiliz $(a + b + 1) \times b$ pisu gutxiago erabiltzen dira, a $\mathbf{x}^{(t)}$ bektoreak dituen elementu kopurua eta b $\mathbf{h}^{(t)}$ bektoreak dituena izanik.

2017. urtera arte *LSTM* eta *GRU* gelaxkekin osatzen diren neurona-sare errepikakorrek artearen egoera definitu zuten hizkuntzaren prozesamenduko hainbat atazetan, *itzulpen automatikoan*, besteak beste. Baina hori orain dela bi urte aldatu zen, *Transformer*ren arrakastak eraginda.

Transformerrak

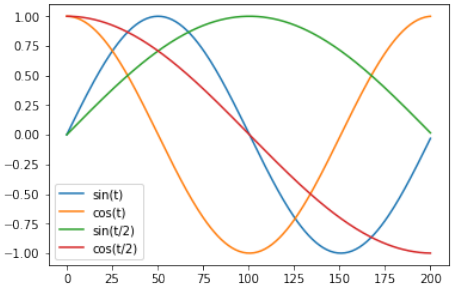
*Transformer*ra aurrerantz-elikatutako neurona-sare konplexu mota bat da [11], bere errendimendua *atentzio-mekanismoen* bidez hobetzen duelarik. Neurona-sare honen azalpena [12] artikuluko irudi eta deskribapenez lagunduta burutu da, neurona-sare honen geruza ezberdinak sakonki deskribatzen ditu eta.

Neurona-sare hauek ataza ezberdinetara moldatzeko flexibilitatea dute, hots, *Transformer*ra sarearen azkeneko geruzak aldatuz *STS* edo *vSTS* atazetara erraz moldatu daiteke; baina erabiliko den azalpena itzulpen automatikoan oinarrituko da, sarrerako esaldi bat beste hizkuntza bateko esanahi bereko esaldi batera bihurtuz.



2.9 Irudia: Transformerraren eskema sinplifikatua. Kasu honetan $n = 6$ kodetzaile eta deskodetzaile aurkitzen dira sarean.

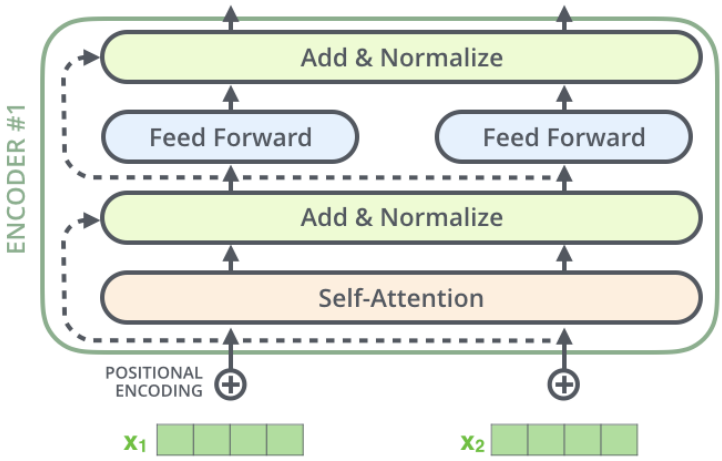
Neurona-sare hau *kodetzaile* eta *deskodetzaile*-multzo bidez osatzen da, deskodetzaile-multzoaren irteeran hainbat geruza ipini daitezkeelarik ebatzi nahi den atazaren arabera. Edozein kasutan, kodetzaile eta deskodetzaile kopurua berdinak dira —2.9 irudia—.



2.10 Irudia: Lau balioz osatutako posizio-kodeketen grafikoa da, $i \in [0, 200]$ kasuetarako definitua.

Sare hau erabiltzeko esaldien d_j elementu dituzten $\mathbf{h}_i$ hitz-bektoreak ikasita eduki behar dira. Hitz-bektore hauei $\mathbf{p}_i$ *posizio-kodeketak* gehitzen zaizkie, hitzek esaldian duten posizioa definitzeko —2.10 irudia—. Behin bi horiek batzean, *Transformer*raren $\mathbf{x}_i$ sarrerako bektoreak lortzen dira. Posizio-kodeketak ezin dira errepikatu i-ren balio ezberdinetarako eta hitz-bektoreen dimentsio bera dute.

Kodetzaile guztien sarrera eta irteera-bektore kopuru eta dimentsioak berdinak dira; eta egitura bereko hainbat kodetzaile badaude ere, kodetzaileen artean ez dira pisuak konpartitzen. Kodetzaile bakoitzak bi azpiatal edo azpi-geruza ditu: *self-attention* edo norberekiko atentzio-geruza, eta geruza-dentso bat —2.11 irudia—. Geruza-dentsoa lehen aipatu den aurrerantz-elikatutako neurona-sarearen parekoa denez, *self-attention* geruza aztertuko da nagusiki.



2.11 Irudia: Sare honek duen kodetzailea. Irudia sinplifikatzeko bi sarrerako bektore azaltzen dira bakarrik, baina sare hauek edozein luzerako esaldiak prozesatu ditzakete sare barruan aldaketarik egin gabe.

Hitz jakin bat kodetzerako orduan, lehenengo azpi-geruza honek esaldi bereko beste hitzei erreparatzea ahalbidetzen du. Izenordainen esanahia zein den aurkitzeko erabilgarria da hau, adibidez. Geruza honen sarrera-bektoretik hiru bektore ezberdin lortzen dira; 2.14 ekuazioetan, $\mathbf{q}_i$ *Query*, $\mathbf{k}_i$ *Key* eta $\mathbf{v}_i$ *Value* i hitzaren bektoreak izanik, hurrenez hurren.

$$\mathbf{q}_i = \mathbf{W_Q} \cdot \mathbf{x}_i + \mathbf{b_Q} \tag{2.14a}$$

$$\mathbf{k}_i = \mathbf{W_K} \cdot \mathbf{x}_i + \mathbf{b_K} \tag{2.14b}$$

$$\mathbf{v}_i = \mathbf{W_V} \cdot \mathbf{x}_i + \mathbf{b_V} \tag{2.14c}$$

Atentzio-geruza honetan h hitz bakoitzak esaldiko hitz guztiekin —esaldiaren hitz kopurua l izanik— zer erlazio duen jakiteko puntuazioak kalkulatzen dira. Puntuazio horiek h hitzaren *Query* bektorea eta aztertu nahi den i hitzaren *Key* bektorearekin kalkulatzen da, bi bektoreen arteko *biderketa-eskalarra* eginez eta $\sqrt{d_k}$ balioarekin zatituz —2.15 ekuazioa—. Azkeneko zatiketa hau sortzen diren gradienteak kontrolpean izateko burutzen da —*exploding-gradient problem* ekiditeko—, d_k balioa *Query*, *Key* eta *Value* bektoreen elementu kopurua delarik.

$$\text{score}_{hi} = \frac{\mathbf{q}_h \cdot \mathbf{k}_i^T}{\sqrt{d_k}} \tag{2.15}$$

Behin h hitzarekin lortu ahal diren score_{hi} puntuazio guztiak kalkulatzen direnean, $\mathbf{score}_h$ puntuazioen artean *Softmax* funtzioa —2.2 ekuazioa— aplikatzen da puntuazioak normalizatzeko, beraien arteko batura batekoa izanik.

Azkenik, s_{hi} puntuazio normalizatuak geruza honen $\tilde{\mathbf{z}}_h$ irteerako bektorea kalkulatzeko erabiliko da, 2.16 ekuazioa jarraituz.

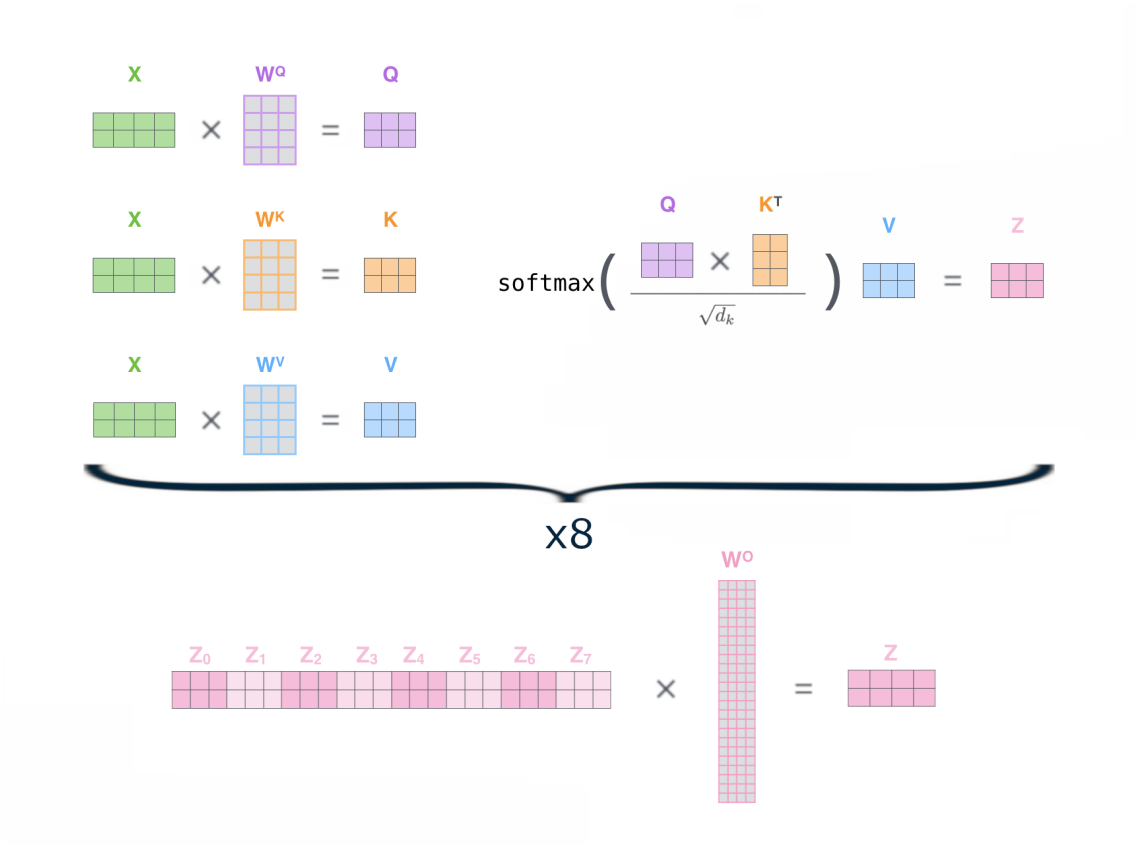
$$\tilde{\mathbf{z}}_h = \sum_{i=0}^{l-1} s_{hi} \cdot \mathbf{v}_i \tag{2.16}$$

Self-attention geruza bakoitzean $\tilde{\mathbf{z}}_h$ behin bakarrik kalkulatu beharrean k aldiz kalkulatzen dira. Ondoren, k $\tilde{\mathbf{z}}_h$ ezberdin horiek konkatenatu eta 2.17 ekuazioa burutzen da, $\mathbf{z}_h$ kalkulatuz. Prozesu honi *Multi-headed attention* deritzo, eta $\tilde{\mathbf{z}}_h$ behin bakarrik kalkulatzeak baino emaitza sendoagoak lortzen ditu.

$$\mathbf{z}_h = \mathbf{W_O} \cdot \left(\tilde{\mathbf{z}}_h^{(0)}, \tilde{\mathbf{z}}_h^{(1)}, \ldots, \tilde{\mathbf{z}}_h^{(k-1)}\right) + \mathbf{b_O} \tag{2.17}$$

Laburbilduz, $\mathbf{x}_h$ sarrera-bektore bakoitzetik *Query*, *Key* eta *Value* bektoreak kalkulatzen dira, horiekin kalkulatu diren puntuazioak erabiliz $\mathbf{z}_h$ irteera-bektorea kalkulatzen delarik. 2.12 irudian *Self-attention* geruzako prozesu osoa irudikatzen da.

Esan bezala, kodetzaile bakoitzean pisuen matrize ezberdinak erabiltzen dira. Beraz, *Self-attention* geruza bakoitzean $\mathbf{W_Q}$, $\mathbf{W_K}$, $\mathbf{W_V}$, $\mathbf{W_O}$, $\mathbf{b_Q}$, $\mathbf{b_K}$, $\mathbf{b_V}$ eta $\mathbf{b_O}$ pisuak ikasi behar dira. Lehenengo hiru matrizeek $d_j \times d_k$ dimentsioa dute. Laugarrenak, berriz, $k \cdot d_k \times d_j$; eta, azkenik, $\mathbf{b}$ bektoreek $3d_k + d_j$ elementu dituzte guztira.



2.12 Irudia: Transformerren *Self-attention* geruzako eragiketak azaltzen dira, $\tilde{\mathbf{z}}_h$ bektoreak $k = 8$ aldiz kalkulatuz.

Beraz, guztira, $(k+3) \cdot d_j \cdot d_k + 3d_j + d_k$ elementu ikasi behar dira *Self-attention* geruza bakoitzeko.

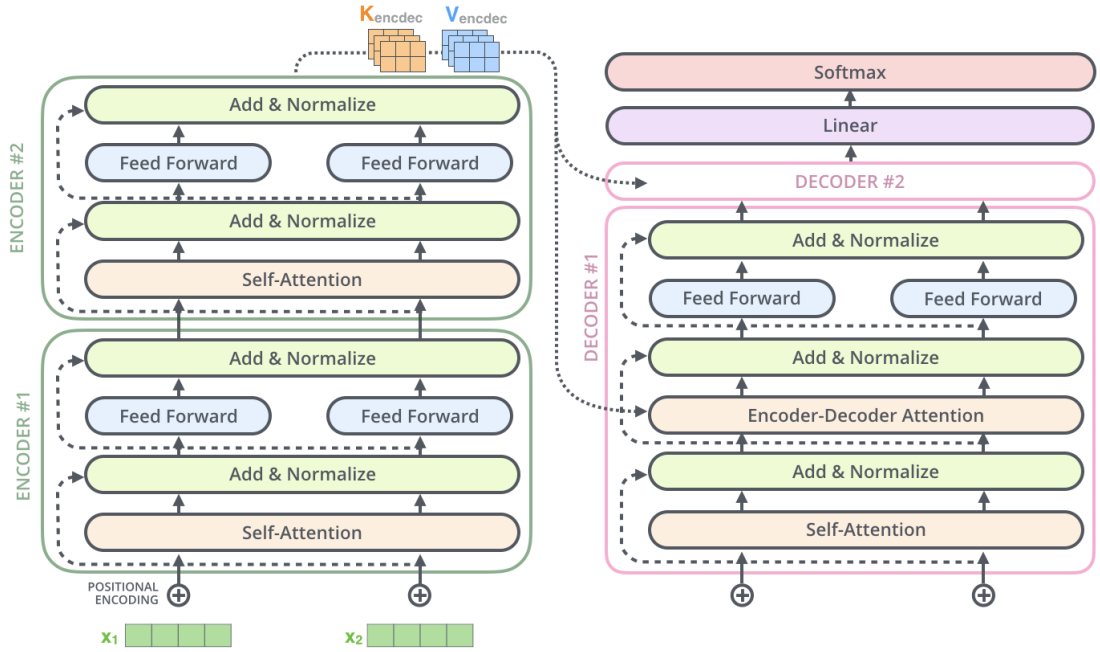
Ondoren, hitz-bektore bakoitza geruza-dentso batetik pasatzen da, sarrera eta irteera-bektore hauek d_j elementu dituztelarik. Hitz guztiek sare berdina erabiltzen dute; beraz, $(d_j+1)d_j$ elementu bakarrik ikasi behar dira geruza honetan —$\mathbf{W_O}$ eta $\mathbf{b_O}$-ren elementuak, 2.18 ekuazioan azaltzen direnak—.

$$\mathbf{o}_h = \mathbf{W_O} \cdot \mathbf{z_h} + \mathbf{b_O} \tag{2.18}$$

Jada ikus daitekeenez, *Transformer*raren kodetzaileek aurrerantz-elikatutako neurona-sare eta neurona-sare errepikakorren geruza batek baino pisu gehiago dituzte —dimentsio bereko hitz-bektoreak erabiltzen badira behintzat—. Ikasi beharreko aldagai kopurua oso handia denez, *Transformer*rekin lan egin denean aurre-entrenatutako ereduak erabili dira, ikasketa-prozesu hauek kostu konputazional handia dakar eta.

2.11 irudian ikus daitekenez, kodetzaileen azpi-geruza guztien ondoren *Add & Normalize izeneko* normalizazio-geruza bat azaltzen da [13]. Eragiketa horretan aurreko azpiataleko geruzako sarrera eta irteera bektoreak batu eta lortzen den emaitza normalizatzen da, ikasketa-prozesua azkartzeko ahaleginean. Deskodetzaileetan berdina egiten da, 2.13 irudian ikus daitekeen moduan.

Deskodetzaileak aldez-aurretik kalkulatu dituen irteerako bektoreak erabiltzen ditu sarrera bektore bezala —hasieran sarrera-bektorerik ez dituelarik—. Hori dela eta, kodetzaile-multzoak aurreranzko propagazio bakar bat behar badu ere, deskodetzaile-multzoak *m* aurreranzko propagazio behar ditu, *m* elementua *Transformer*rak bukaeran sortzen duen esaldiaren hitz kopurua izanik. Aurreranzko propagazio bakoitzeko esaldiaren hurrengo hitza estimatzen du deskodetzaileak, esaldia bukatzen delaren sinbolo berezi bat estimatu arte. Horrela, sarrerako esaldiaren hitz kopuru ezberdinak dituen irteerako esaldia sortu dezake.



2.13 Irudia: Bi kodetzaile eta deskodetzailez osatutako *Transformerra*. Behin bigarren kodetzailearen irteera-bektoreak lortzean, *Key* eta *Value* bektoreak kalkulatzen dira, bektore hauek deskodetzaileko *Encoder-Decoder Attention* azpi-geruzan erabiltzen direlarik. Bukaeran, geruza-dentso bat dago *Softmax* aktibazio-funtzioarekin, sarearen bukaerako estimazioa burutzen duena.

Deskodetzaileen kasuan, kodetzailetan aurki daitzeken *Self-attention* eta *Feedforward* geruzetaz gain beste azpi-geruza bat aurki daiteke, bi horien artean dagoena. *Encoder-Decoder Attention* geruza honek kodetzaile-multzo bukaeran lortzen diren irteera-bektoreen *Key* eta *Value* eta deskodetzailearen aurreko azpi-geruzan kalkulatutako bektoreen

Query bektoreak erabiltzen ditu, deskodetzaileek kodetzaile-multzoak ikasi dituen ezaugarrietan atentzioa jarri dezan —2.13 irudia—.

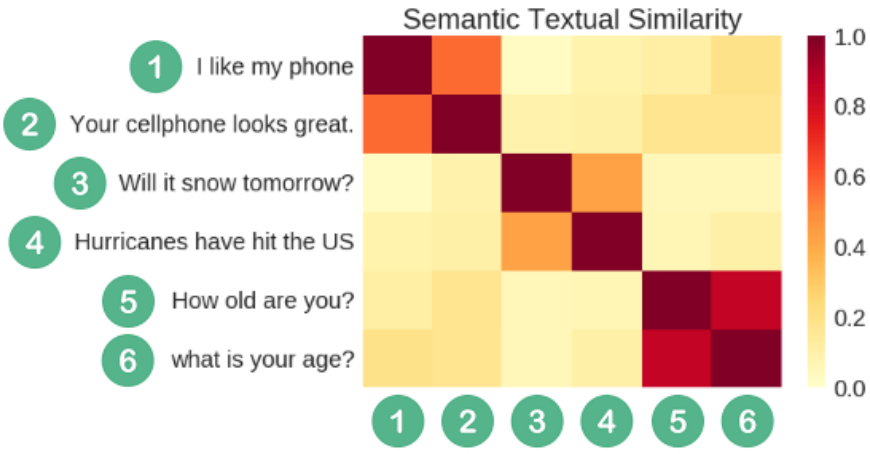
Azkenik, aipatzekoa da deskodetzailetako *Self-attention* geruza ez dela kodetzailearen guztiz berbera; deskodetzaile-multzoak estimatuko duen hurrengo hitza kalkulatzeko, jada estimatu diren hitz-bektoreak aztertu ditzake bakarrik eta.

2.1.2 Erabili diren sistemak

Proiektu honetan testuen errepresentazioak lortzeko lau *embedding* mota ezberdin erabili dira. Horietatik, *Universal Sentence Encoder* saretik lortutako esaldi-bektoreak erabili dira; *Bidirectional Encoded Representations from Transformers* eta *Generative Pretrained Transformer* sareetatik eratorritako hitz-bektoreak erabili dira ere bai; eta, azkenik, *Global Vectors* hitz-bektoreak erabili dira, hitzen arteko *agerkidetza* aztertuz sortu direnak. Hitz-bektoreetatik esaldi-bektoreak lortzeko hauen batez-bestekoak erabili dira.

Universal Sentence Encoder

Universal Sentence Encoder edo *USE* esaldi-bektore orokorrak lortzeko diseinatu den eredua da [14]. Kodetzaile batez osatuta dago, eta ahalik eta adierazpen orokorrenak lortzeko helburuarekin sortu da. Hau ereduaren ikasketa-prozesua hainbat atazetan zehar eginik lortzen da, *STS* ataza horietako bat delarik —2.14 irudia—.

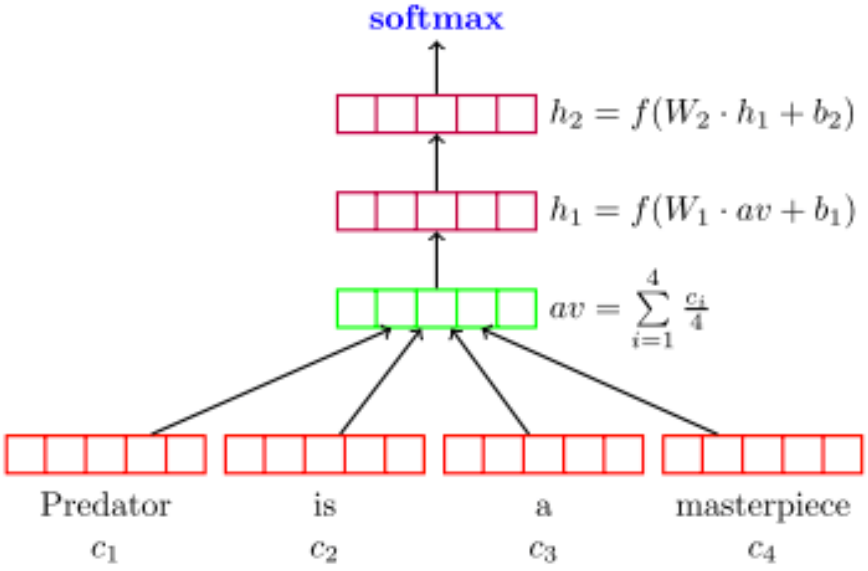


2.14 Irudia: *USE-D* ereduak ematen dituen esaldi-bektore pareen arteko antzekotasun semantikoak kalkulatu dira, bien arteko *kosinuaren-antzekotasuna* erabiliz. Zero balioak bi esaldiek antzekotasunik ez dutela adierazten du; eta batekoak, berriz, semantikoki parekoak direla [14].

Berez, bi *USE* ezberdin daude. Alde batetik, *USE-T* eredua dago, *Transformer* arkitekturan oinarritua dagoena. Zehaztasun handiko emaitzak itzultzen ditu, baina kostu-konputazional handia dakar *USE-T* entrenatzeak —ez da esaldien luzerarekiko linealki proportzionala—. Beste aldetik, *USE-D* eredua, *DAN* deritzon aurrerantz-elikatutako neurona-sare mota batean oinarrituta dago [15]. Lortzen dituen emaitzak *USE-T* sistemaren parekoak dira —atazaren arabera honen emaitzak hobetzen dituelarik—, eta *USE-D*-ren kostu-konputazionala esaldiaren luzerarekiko linealki proportzionala da.

Lan honetan *USE-D* erabili da bakarrik. Hori dela eta, hemendik aurrera *USE-D*-ri buruz hitz egitean *USE* akronimoa erabiliko da.

Hitz gutxitan, *USE* eredu hau *DAN* edo *Deep Averaging Network* sisteman oinarritzen da. *DAN* aurrerantz-elikatutako neurona-sare sinple bat da; baina, lehenengo geruzako eragiketak egin baino lehen, sarrera-bektore guztien arteko batez-bestekoa kalkulatzen da. Adibidea 2.15 irudian aurki daiteke.



2.15 Irudia: *DAN* neurona-sarearen egitura. Sarrera-bektoreen arteko batez-bestekoa kalkulatu ondoren, bi geruza-dentso agertzen dira, baina geruza kopurua sarearen arabera aldatzen da —ez da *USE* ereduak erabiltzen dituen geruza kopurua— [15].

Esaldi-bektorea lortzeko prozesua honakoa da. Lehenik eta behin, esaldiari letra larriak xehetan bihurtzen dira. Ondoren, esaldia tokenizatu egiten da. Tokenizazio berezi bat erabiltzen da, *PTB* —edo *Penn Treebank 3*— deiturikoa; eta token bakoitzaren bektoreak lortzeko berriez osatutako datu multzo handi batean aurre-entrenatutako *word2vec* eredu

bat erabiltzen du [16]. Behin hitz-bektoreak —edo hobe esanda, token-bektoreak— izanik, 2.15 irudiko eskema jarraitzea besterik ez da geratzen. Lortzen den esaldi-bektorea 512 balioz osatuta dago, eta *STS* atazan ondo moldatzen da bi esaldi-bektoreen arteko kosinuaren-antzekotasuna erabiliz.

Global Vectors

Bektore globalak —edo *GloVe* bektoreak— hitzen arteko agerkidetzen probabilitateak erabiliz kalkulatzen dira [1], probabilitate horiek hitzen esanahia kodetzeko informazioa dutela kontuan harturik. Sistema log-bilinear bat erabiltzen da *GloVe* bektoreen ikasketan, eta honen helburua hitz-bektoreen arteko biderketa eskalarrak aipatutako probabilitatearen logaritmoa ematea da, 2.19 ekuazioan azaltzen den bezala.

$$\mathbf{w}_i^T \cdot \tilde{\mathbf{w}}_j + \mathbf{b}_i + \tilde{\mathbf{b}}_j = \log(\mathbf{X}_{ij}) \tag{2.19}$$

Kontuan izanik bi balioen zatiketa haien arteko logaritmoen diferentzia dela, helburu honek agerkidetza ratioen logaritmoak hitz-bektoreen arteko diferentziekin erlazionatzen ditu, gertu dauden hitz-bektoreak antzekotasun semantiko sakon bat izanik —2.16 irudia— eta hitzen arteko analogiak eraiki ahal izanik —1.1 irudian aipatzen dena—.

2.16 Irudia: GloVe hitz-bektoreak erabiliz *frog* hitzaren zazpi hitz-bektore gertuenak azaltzen dira ordenaturik, *k-NN* edo *k-Nearest Neighbor* algoritmoa balitz bezala —$k = 7$ izanik— [1].

2.19 ekuazioko notazioari begira X matrizeak hitzen arteko agerkidetza matrizea definitzen du, non X_{ij} adierazpenak j hitzak i hitzaren testuinguruan dituen agerpen kopurua definitzen duen, V erabiltzen den hitz guztien multzoa eta $\forall i, j \in V$ izanik. $\mathbf{w}$, $\tilde{\mathbf{w}}$, $\mathbf{b}$ eta $\tilde{\mathbf{b}}$ ikasi beharreko matrizeak dira, non matrize horien lerro bakoitzak 300 elementuko hitz-bektore bat jasotzen duen, $V \times 300$ dimentsioko matrizeak izanik.

Matrize horiek ikastean $X_{ij} \neq 0$ kasuak bakarrik hartzen dira kontuan, besteak erabiltzen ez direlarik. X simetrikoa denez, $\mathbf{w}$ eta $\tilde{\mathbf{w}}$ ezberdinak erabiltzea ez litzateke beharko, matrize hauen ausazko hasieraketak bakarrik eragingo baititu bien arteko aldaketak. Hala

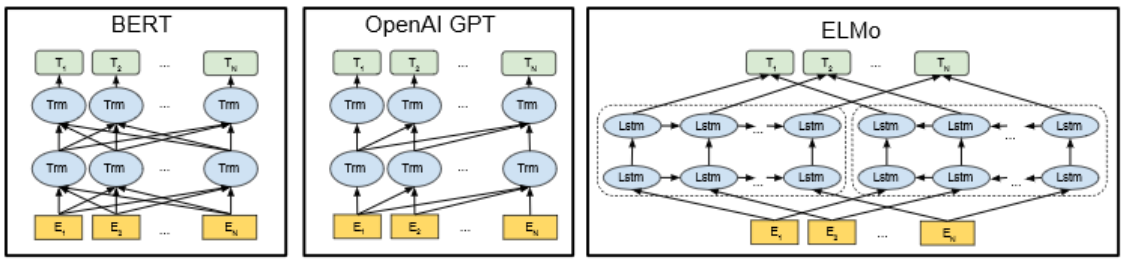
ere, *GloVe*-ren egileek bi matrize ezberdinak erabiltzea gomendatzen dute, hitz-bektore hauen errendimendua hobetzeko intentzioarekin —normalean, sistemaren instantzia anitzak edukitzeak lortzen diren emaitzak hobetzea dakar eta— [17]. i hitzaren $\mathbf{g}_i$ *GloVe* hitz-bektorea 2.20 ekuazioaren bidez lortzen da —$\mathbf{b}$ eta $\tilde{\mathbf{b}}$ ez dira kontuan hartzen bektore hauen sorkuntzarako—.

$$\mathbf{g}_i = \mathbf{w}_i + \tilde{\mathbf{w}}_i \tag{2.20}$$

Bidirectional Encoded Representations from Transformers

Konputagailu bidezko ikusmenean transferentzia ikasketa —hots, sistema jakin bat guztiz ezberdina den ataza batean ebaluatzea— asko erabili da. Hala ere, azken urteotan bere erabilera hizkuntzaren-prozesamendurako egokia dela ikusi da.

BERT edo *Bidirectional Encoded Representations from Transformers* sistemak hizkuntzaren-prozesamenduko hainbat atazetan artearen egoerako emaitzak lortu ditu ikasketa-transferentzia erabiliz [18]. *Google*-n sistema honen gakoa hizkuntza-eredu bat eraikitzeko *bi noranzkoko Transformer* baten erabilera da. *BERT* arkitektura sortu aurretik, testu sekuentziak ezkerretik eskuinera aztertu, edota ezkerretik eskuinera eta eskuinetik ezkerrerako ikasketak konbinatzen zituzten. *BERT*ek, ordea, bi noranzkoak ikasketa berean konbinatzen ditu; *Masked LM* edo *MLM*, eta *Next Sequence Prediction* edo *NSP* teknikek ahalbidetzen dutelarik —2.17 irudia—.



2.17 Irudia: Hiru arkitektura aurki daitezke [18]. Ezkerretik eskuinera: *BERT*, *GPT* eta *ELMo*.

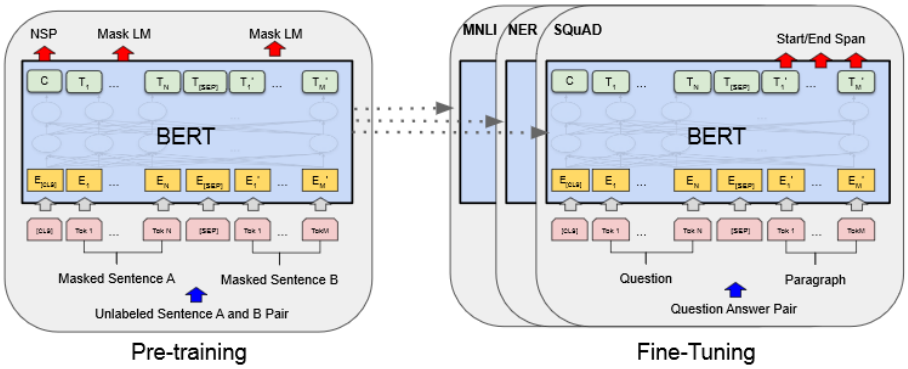
Esan bezala, 2.17 irudiko bigarren sistema *OpenAI* enpresaren *Generative Pretrained Transformer* edo *GPT* da [19], ezkerretik eskuinerako irakurketa burutzen duen *Transformer*ra dena. Hirugarrena, aldiz, *ELMo* deitzen da [20]. *ELMo*-k *LSTM*-ak erabiltzen ditu, testu sekuentzien ezkerretik eskuinera eta eskuinetik ezkerrerako irakurketa sekuentzialak konbinatuz.

Bi sare hauek esaldia prozesatzean norabide jakin bat jarraitzen dute, baina hau ez da *BERT* sarearen kasua. *BERT*-en kodetzaileek esaldien hitz guztiak batera irakurtzen dituzte. Hortaz, *BERT-ek* bi noranzko dituela esan ohi da, noranzkorik ez duela esatea zehatzagoa bada ere —hitz baten kontextua bere inguruan dauden hitzekin ikasten delarik—.

BERT sistemak 30.000 hitzeko hiztegia erabiltzen du, *WordPiece* hitz-bektoreak erabiliz [21]. Hau horrela izanik, lehenengo sarrerako esaldiak tokenizatu behar ditu. Tokenizazioa eginda dagoenean, token berezi batzuk gehitu behar dira. Lehenengo tokena [CLS] tokena izango da beti. Esaldiaren bukaeran, berriz, [SEP] tokena jarri behar da, esaldia bukatzen dela adierazten duelarik. Sistema honek bi esaldi batera sartzeko aukera ematen duenez, lehenengo eta bigarren esaldi artean beste [SEP] token bat jarri behar da —ondoren ikusten den adibidean bezala—.

```
[CLS] Somebody once told me the world is gonna roll me [SEP]
[CLS] My Name is Ander [SEP] I am 23 years old [SEP]
```

Behin token guztiak prest daudela, hitz-bektore horiek posizio-bektoreekin batu eta 12 kodetzailez[1] osatutako *Transformer*raren kodetzaile-multzotik pasatzen dira. Lehen azaldu den oinarrizko *Transformer*rean ez bezala, *BERT* sistemak ez du deskodetzaile-multzorik, hizkuntza-eredu bat ikasteko ez baita beharrezkoa. Beraz, proiektu honetan erabiliko diren adierazpenak kodetzaile-multzotik lortzen direnak izango dira.



2.18 Irudia: Irudi honetan *BERT* sistemaren bi ikasketa-faseak ikusi daitezke. Lehenengoan *NSP* eta *MLM* atazak erabiltzen dira hizkuntza-eredu bat eraikitzeko ahaleginean. Bigarrenean, berriz, nahi den ataza ebazteko ikasketa burutzen da —berfintzea—, bigarren fase honetan egin behar diren aldaketak ezberdinak direlarik atazaren arabera [18].

[1] *BERT-base* edo *BERT-large* bertsioen arabera 12 edo 24 kodetzaile izango dira. 12 kodetzaileko bertsioaren emaitzak ez dira bestearenak bezain onak, baina errendimendu ona ematen du ikasketa-prozesu murriztuago batekin. Hori dela eta, *BERT-base* bertsioa erabili da esperimentuetan.

Esan bezala, *Transformer* hau ikasteko orduan bi ataza ezberdin erabili dira. Alde batetik, *Masked LM* ataza erabili da, hizkuntza-eredu baten pareko ataza burutzen duena. Ataza honetan tokenen %15-ari [MASK] tokena ezartzen zaie, sarearen irteeran token horiek zeintzuk diren aurreikusi behar direlarik. Teknika honek aurretik erabili ezin ziren bi noranzkoko *Transformer*rak ikastea ahalbidetzen du.

Beste aldetik, *NSP* atazan helburua bigarren esaldia lehenengoaren hurrengoa ala ez den asmatzea da. Asmatze hori [CLS] tokenaren kodetzaile-multzoaren irteeraren gainean geruza-dentso bat gehiago jarriz egin daiteke, CLS geruza deritzona —2.18 irudia—. Bi ataza hauekin testu soila erabiliz aurre-entrenatzen da *BERT* —anotatu gabeko testuarekin, hain zuzen ere—. Ondoren, CLS geruza aldatu edo berriz entrenatu daiteke —aurretik ikasitako pisuak erabiliz— ataza ezberdinak ebatzi ahal izateko —STS ataza, adibidez—, transferentzia ikasketa erabiliz eta *berfinketa* edo *fine-tuning* bat eginik.

Erabilitako *BERT* sistemaren oinarrizko bertsioak 12 kodetzaile erabiltzen ditu. Hitzbektore bakoitza 768 elementuz osatzen da, eta kodetzaile bakoitzean 12 buruko atentzio-mekanismoak erabili dira —*Multi-headed attention* delakoa erabiliz—.

Proiektu hau bukatu baino bi hilabete lehenago, *Google*k *Transformer*retan oinarrituta dagoen *XLNet* sistema berri bat publikatu zuen [22], artearen egoera berria definituz hizkuntzaren-prozesamenduko hainbat atazetan —*STS barne*—. Hala ere, ordurako proba gehienak eginak zeudenez, sistema berri hau ez da erabili proiektuko esperimentuetan. Hizkuntzaren prozesamenduan emaitzak etengabe ari dira hobetzen. Izan ere, *BERT* eta *XLNet* sistemen publikazioen artean 9 hilabete bakarrik pasa ziren.

Generative Pretrained Transformer - 2

GPT-2 sistemak lehen azaldutako *GPT*-ren pareko arkitektura du [23]. Funtsean, bi arkitektura hauek aurrez aldetik aipatutako oinarrizko *Transformer* baten deskodetzaile-multzoa erabiltzen dute, eta, gehienbat, hizkuntzaren-sorkuntzarako erabiltzen dira. 2.17 irudian azaltzen den bezala, *GPT* sistemak ezker-eskuinerako atentzio-mekanismoak ditu bakarrik, *BERT*en sorkuntzan oinarritzat hartu zela esan ohi delarik.

*BERT*ek bezala, *GPT* familia osatzen duten bi sistemek ikasketa-prozesu bera dute. Lehenengo fasean, aurre-ikasketa deritzona, hizkuntza-eredu tradizional bat ikasten da, esaldi baten hurrengo hitza estimatzen saiatzen dena. Bigarren fasean, berriz, berfintze bat aplikatzen zaio arkitekturari, atazaren arabera aldatzen dena. *GPT* sistemek *BERT*en aipatzen

diren [CLS] eta [SEP] token bereziak ere erabiltzen dituzte; baina, *BERT* arkitekturan ez bezala, berfintze-prozesuan erabiltzen dira bakarrik.

Tokenei buruz hitz egiten ari garela aprobetxatuz, tokenizazio berezi bat erabiltzen dute *GPT* familiako sistemek, *BPE* edo *Byte Pair Encoding* [24], non tokenizazio hori *corpus* batetik ikasten den. Tokenizazioaren ikasketaren hasieran hiztegia alfabetoak dituen letrez osatuta dago, eta *corpus* horretan maiz azaltzen diren letra segidak hiztegira gehitzen doaz, *corpus*eko hitz eta silaba esanguratsuenak hiztegira sartzen direlarik. Ikasketa ondoren ezagutzen ez diren tokenak zatitu egiten dira ezagutzen diren tokenak lortu arte.

GPT eta *GPT-2* arteko ezberdintasun nagusia beraien arteko tamaina diferentzia da. Azkenekoak lehenengoak baino parametro gehiago erabiltzen ditu —10 aldiz gehiago—. Guztira, *GPT-2* sistemak ikasi beharreko $1.5 \cdot 10^9$ parametro ditu. Gainera, hiztegiaren tamaina handiagoa erabiltzen du *GPT-2* bertsioak, eta 2.13 irudiko deskodetzaileko normalizazio-geruzak atentzio-geruzen ondoren aplikatu beharrean, aurretik aplikatzen dira.

Lehen esan bezala, sistema hauek testua sortzeko erabiltzen dira eta 2.1 taulan azaltzen den testua sor dezake, adibidez.

In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.
The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science. Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved. Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a tiny valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow. Pérez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Pérez. Pérez and his friends were astonished to see the unicorn herd. These creatures could be seen from the air without having to move too much to see them – they were so close they could touch their horns. While examining these bizarre creatures the scientists discovered that the creatures also spoke some fairly regular English. Pérez stated, "We can see, for example, that they have a common 'language,' something like a dialect or dialectic."

2.1 Taula: Taula honetan *GPT-2* sistemak sortu duen testua azaltzen da, letra lodiz idatzitakoa gizaki batek idatzi duelarik, sistemaren sarrerako testua izanik.

Hizkuntzaren-sorkuntzan oinarritzen ez diren atazetan, hots, *STS* bezalako atazetan, emai-

tzak ez dira *BERT* sistemakoak bezain onak —antzekoak badira ere—; baina, hala ere, esperimentu ez-gainbegiratuetan *GPT-2*-ren irteera-bektoreak erabili dira, antzekotasun semantikoa aztertzeko egokiak izan daitezke eta.

2.1.3 Datu multzoak

Lan honetan hiru datu multzo ezberdin erabili dira. Horietako bi, bi esaldiren arteko antzekotasun semantikoak aztertzeko eraiki dira, *STS* eta *vSTS* atazak ebazteko alegia; eta azkena, berriz, datu base orokorrago bat da, 4. kapituluan azalduko den arkitektura berria entrenatzeko orduan erabiliko dena.

5	*Bi esaldiak guztiz parekoak dira, esanahi bera dute eta.*
	The bird is bathing in the sink. Birdie is washing itself in the water basin.
4	*Bi esaldiak ia parekoak dira, baina hainbat detaile ezberdin dituzte.*
	Two boys on a couch are playing video games. The boys are playing a video game.
3	*Bi esaldiak nahiko antzekoak dira, baina hainbat ezaugarri garrantzitsu ezberdin dituzte.*
	John said he is considered a witness but not a suspect. "He is not a suspect anymore". John said.
2	*Bi esaldiak ez dira parekoak, baina ezaugarri batzuk partekatzen dituzte.*
	They flew out of the nest in groups. They flew into the nest together.
1	*Bi esaldiak ez dira parekoak, baina gai bera partekatzen dute.*
	The woman is playing the violin. The young lady enjoys listening to the guitar.
0	*Bi esaldiak guztiz ezberdinak dira.*
	The black dog is running through the snow. A race car driver is driving his car thorugh the mud.

2.2 Taula: Antzekotasun semantikoa definitzen duten balio batzuen adibideak, balio horien esanahiekin batera.

STS eta *vSTS* atazetan antzekotasun semantikoa 0 eta 5 arteko balioekin neurtzen da, non, bi esaldi izanik, 0 batek bien arteko antzekotasun semantikorik ez dagoela esan nahi duen. 5 batek, aldiz, bi esaldiek esanahi berdina dutela adierazten du. 2.2 taulan, balio horien definizio zehatzagoak aurki daitezke.

Hala ere, hiruetatik batean bakarrik lantzen dira esklusiboki testuak —besteetan irudiak ere azaltzen baitira—, *STS-Benchmark* datu multzoan hain zuzen ere.

STS-Benchmark

Datu multzo hau 2012 eta 2017 urteetan zehar *SemEval* [3] atazetan erabili diren datu multzoetako hainbat instantzien bidez eraiki da, instantzia bat esaldi pare bat gehi horien antzekotasun semantikoa definitzen duen balioa izanik. *STS* ebazten duten sistema berriak konparatu ahal izateko sortu da, estandarra bihurtu dena; eta, guztira, 8628 instantzia ezberdin ditu.

Aurki daitezkeen instantziak hiru azpimultzo disjuntutan banatu dira: *train*, *dev* eta *test*. Lehenengoa, azpimultzo handiena, ataza ebatzi behar duten sistemak entrenatzeko definitu da, hots, ikasketa-prozesuan erabiltzeko. Bigarrena, berriz, sistema horien berfintze edo *fine-tuning* egiteko dago pentsatua. Azkenik, *test* azpimultzoa afinatu berri den sistemaren orokortze gaitasuna aztertzeko erabiltzen da, hots, sistemak momentu horretara arte ikusi ez dituen instantziekin duen errendimendua ikusi eta beste sistemen errendimenduarekin konparatzeko erabiltzen da.

Gainera, datu multzoan aurki daitezkeen testuak barietate handikoak eta iturburu ezberdinekoak dira. Esaldi-pare batzuk irudien goiburukoak dira; eta, beste batzuk, berriz, berrien izenburuetatik edota hainbat foroetatik eskuratu dira, datu multzoak 2.3 taulan agertzen den distribuzioa duelarik.

	Train	**Dev**	**Test**	**Guztira**	**Train**	**Dev**	**Test**	**Guztira**
Berriak	3299	500	500	4299	%38.24	%5.79	%5.79	%49.83
Goiburuko.	2000	625	625	3250	%23.18	%7.24	%7.24	%37.67
Foroak	450	375	254	1079	%5.21	%4.34	%2.94	%12.5
Guztira	5749	1500	1379	8628	%66.64	%17.38	%15.98	%100.0

2.3 Taula: Taula hauetan azpimultzo bakoitzak duen instantzia kopuru eta proportzioak aurki daitezke, hurrenez hurren. Instantzia bakoitzak bi esaldi ditu; beraz, esaldi kopurua tauletako balioen bikoitza da.

Foroetatik datozen instantziek multzoaren zortzirena bakarrik osatzen dute, baina zortziren horrek *STS* ebazteko azpiatal zailena osatzen du. Izan ere, normalean, goiburuko eta berrietako esaldiek antzeko egitura sintaktikoak jarraitzen dituzte, eta foroetako instantzietan daudenak, berriz, ez dute egitura finkorik jarraitzen.

Hitz-zakuak testuak adierazteko modu bat da, non testu jakin bateko hitz guztiak jasotzen diren, hauen agerpen kopuruarekin batera. 2.4 taulan *STS-B* datu multzoko *News*, *Cap-*

tions eta *Forum* azpiataleko hitz-zakuak agertzen dira, azpiatal bakoitzeko ezaugarriak aztertzeko erabilgarria dena.

Hiru azpiatalen barruan hainbat ezberdintasun aurki daitezke. Kasurik nabariena *caption* edo irudien goiburukoena da. Izan ere, azpiatal honetan honako egitura hauek erabiltzen dira ia esaldi guztietan —koloreztatuta dauden hitzak ordezkatu behar direnak dira, eta parentesi artekoak kasu gehienetan agertzen dira—.

A subject (is) verb+(ing) something.
Subjects (are) verb+(ing) something.

News		**Caption**		**Forum**	
All	without SW	All	without SW	All	without SW
the: 0.4919	said: 0.0771	a: 1.3589	man: 0.2532	the: 0.4958	think: 0.0426
in: 0.358	us: 0.0645	is: 0.4606	woman: 0.1562	to: 0.3234	answer: 0.0426
to: 0.3018	percent: 0.0445	the: 0.3843	dog: 0.1058	you: 0.3044	want: 0.0403
of: 0.2326	killed: 0.042	man: 0.2532	playing: 0.1002	a: 0.2827	question: 0.0343
and: 0.1648	new: 0.0397	in: 0.2437	two: 0.0915	is: 0.2748	yes: 0.0334
a: 0.1603	syria: 0.0324	on: 0.1974	white: 0.0785	i: 0.2053	know: 0.0297
on: 0.1178	china: 0.0277	woman: 0.1562	black: 0.0691	of: 0.1872	one: 0.0297
for: 0.1174	police: 0.0265	and: 0.1294	water: 0.0414	it: 0.1715	many: 0.0283
at: 0.0876	president: 0.0241	of: 0.1217	people: 0.0411	in: 0.1627	would: 0.0278
said: 0.0771	two: 0.0217	with: 0.1063	girl: 0.0402	and: 0.1497	depends: 0.0264
us: 0.0645	nuclear: 0.0211	dog: 0.1058	boy: 0.0394	are: 0.1284	problem: 0.0255
was: 0.0628	kills: 0.0211	playing: 0.1002	running: 0.0391	not: 0.1242	need: 0.0246
is: 0.0612	million: 0.0207	two: 0.0915	sitting: 0.0378	that: 0.1214	like: 0.0246
that: 0.0591	pakistan: 0.0201	are: 0.0814	riding: 0.0374	for: 0.1057	things: 0.0222
with: 0.0544	dead: 0.0197	white: 0.0785	standing: 0.0368	have: 0.1057	good: 0.0222
from: 0.0518	iran: 0.0188	black: 0.0691	guitar: 0.0357	do: 0.1033	get: 0.0181
as: 0.0516	talks: 0.0184	an: 0.0508	brown: 0.0345	this: 0.0899	use: 0.0176
by: 0.0452	attack: 0.0174	at: 0.0508	person: 0.0342	what: 0.0862	vessels: 0.0176
percent: 0.0445	first: 0.0172	to: 0.042	dogs: 0.0325	on: 0.0829	filled: 0.0176
killed: 0.042	would: 0.0167	water: 0.0414	cat: 0.0323	your: 0.0811	wonders: 0.0176

2.4 Taula: Zutabe bakoitzaren goiburukoetan taulako *News*, *Caption* eta *Forum* azpiatal bakoitzeko hitz guztiak kontuan hartu ala hitz hutsak —*stop words* edo *SW*— kendu diren ikus daiteke. Zutabe bakoitzean azpiatal horietan gehien errepikatzen diren 20 hitzak azaltzen dira, non dagokien balioa esaldi bakoitzeko batez-besteko agerpen kopurua den.

Hau erraz ikus daiteke 2.4 taulako *caption* azpiataleko hitz-zakuetan. Adibide bat jartzeagatik, *a* hitza batez-beste 1.35 aldiz agertzen dela ikus daiteke esaldi bakoiteko, hitz honen agerpen kopurua oso altua izanik. Egitura bera erabiltzeak erabilitako neurona-sareei ataza errazten die, eta hau esperimentuak hartzean kontuan hartu behar da, *vSTS* datu multzoa

erabiltzean batez ere. *Caption*ek, oro har, egitura bera eta hiztegi errepikakor bat —oso generikoa dena— erabiltzen dute.

News azpimultzoan, berriz, *Syria* eta *China* bezalako izen bereziak maiz agertzen direla ikus daiteke, beste azpiataletan izen bereziak ia erabiltzen ez direlarik. Azkenik, *Forum* azpiatalean *hitz hutsen*[2] erabileran barietate handiagoa antzeman daiteke. Normalean foroetan hizkuntza anonimo eta informal bat erabiltzen da, izenordain eta preposizioz betea. Hauxe erraz antzematen da, azpiatal honetako 20 hitz erabilienak hitz hutsak baitira —beste azpimultzoetan ez bezala—. Beraz, foroetan dauden esaldien esanahi semantikoa prozesatzea beste azpimultzoena baina zailagoa bihurtu daiteke.

Normalean azpimultzo bakoitzeko hitz-zakuen tamaina —edo erabilitako hiztegiaren hitz kopurua— kontuan hartzekoa izango litzateke. Hala ere, 2.3 taulan azpimultzo bakoitzak instantzia kopuru oso ezberdinak dituztela ikus daiteke. Hortaz, kopuru horiek informazio handirik ematen ez duela erabaki da.

Esaldien luzerak hauen konplexutasuna adierazi dezakete ere bai. 2.5 taulan ikusi daitekeenez, *Caption* azpimultzoko esaldiak motzagoak dira, oro har, beste multzokoak baino.

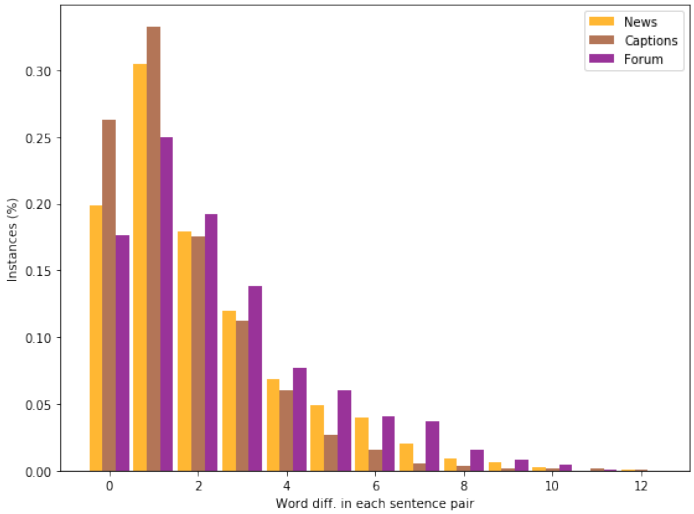
	Mean	**Std**	**Min**	**25%**	**50%**	**75%**	**Max**
News	11.186	6.256	3	7	9	15	55
Caption	8.079	2.866	2	6	7	10	27
Forum	10.636	4.456	2	7	10	14	22

2.5 Taula: STS-B dat u-multzoko esaldien hitz kopuruak, hiru azpimultzoetan banatuta. Ezkerretik eskuinera, hitz kopuruen batez-bestekoa, desbideraketa-estandarra, minimoa, lehen kuartila, mediana, bigarren kuartila eta maximoa.

Beste aldetik, instantzia bereko esaldien luzerak ezberdinak izan daitezke. Hitz kopuru horien arteko diferentziak hainbat sistemen errendimendua jaistea ekar dezake, *Transformer*ren irteera geruzako token kopurua alda dezakete eta, adibidez. Kontuan izan behar da erabiltzen ari den tokenizazioaren arabera token kopuru hori ere aldatzen dela, baina hitz kopuruen diferentziak errealitatean gertatuko diren ezberdintasunen parekoak izango direla suposatzen da —2.19 irudia—.

Datu multzo honen analisiarekin bukatzeko datu multzoak dituen instantzien antzekotasun balioak aztertuko dira. Aurrez aldetik aipatutako sistemak entrenatzerako orduan datu multzoan estimatu nahi diren balioek distribuzio uniforme edo orekatu bat jarraitzea oso garrantzitsua da, hots, 0 eta 5 arteko antzekotasun semantikoak dituzten instantziak uniformeki zabaltzea tarte horretan.

[2]Esanahi semantikorik gabeko hitzak dira: izenordainak, preposizioak, artikuluak, besteak beste.



2.19 Irudia: STS-B datu multzoko esaldi pareen hitz kopuruen diferentziak, hiru azpimultzoetako instantziak ezberdinduz. Ardatz bertikalean azpiatal bakoitzaren proportzioak definitzen dira, eta horizontalean, berriz, instantzia bakoitzaren esaldien hitz kopuru diferentzia.

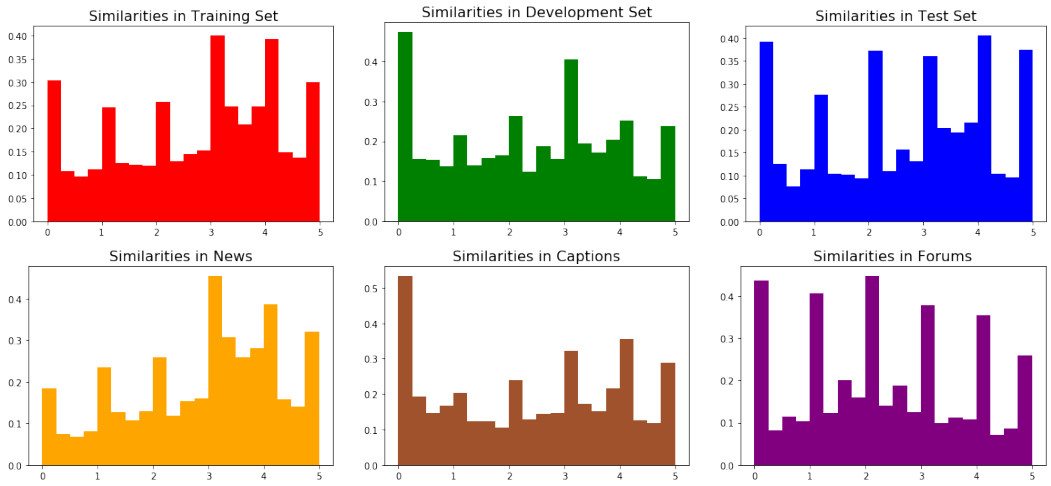
	Mean	**Min**	**25%**	**50%**	**75%**	**Max**	**Zero Kop.**
News	2.884	0	1.8	3.2	4	5	%3.72
Caption	2.392	0	1.0	2.2	3.8	5	%11.01
Forum	2.311	0	0.6	1.8	3.4	5	%9.26
All	2.627	0	1.4	2.8	3.8	5	%7.16

2.6 Taula: *STS-B* datu multzoko instantzien antzekotasun-balioen laburpena, hiru azpimultzoetan banatuta edota osorik. Ezkerretik eskuinera, hitz kopuruen batez-bestekoa, minimoa, lehen kuartila, mediana, bigarren kuartila, maximoa eta zero kopuruak agertzen dira.

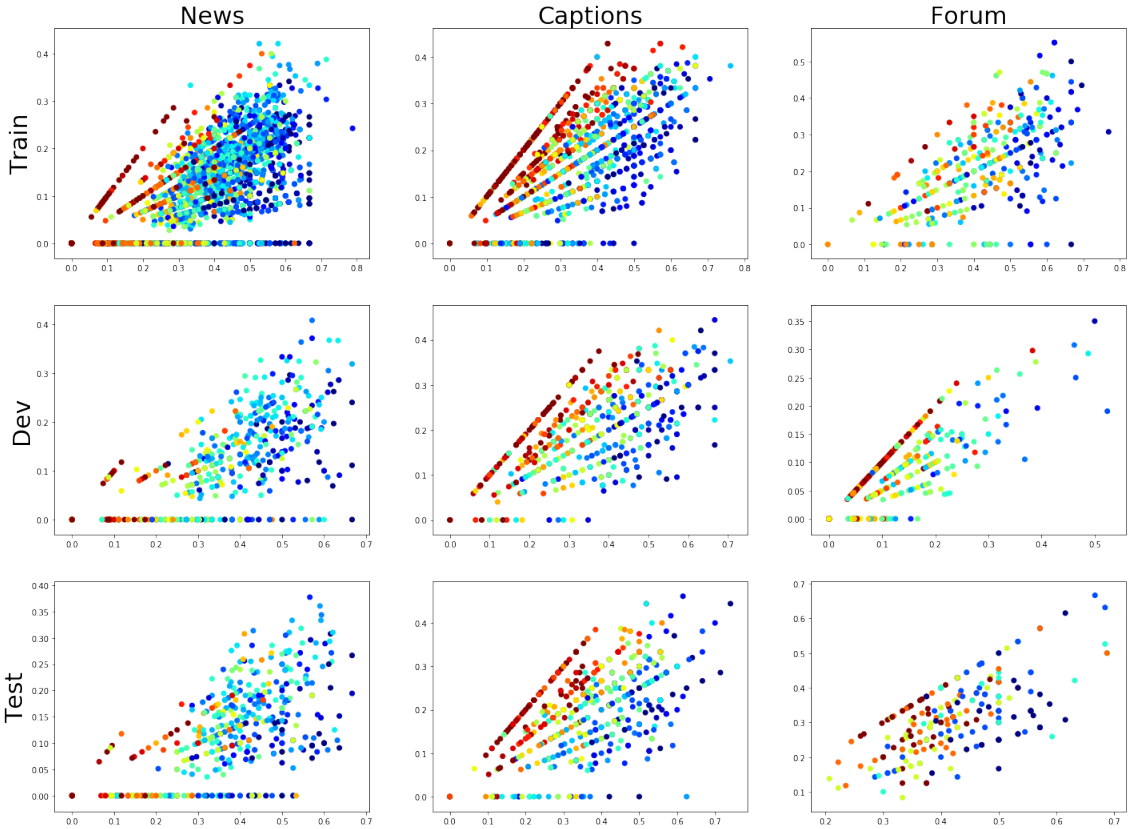
2.6 taulan aldez aurretik definitutako *News*, *Caption* eta *Forum* azpimultzoetan agertzen diren instantzien antzekotasun balioen laburpena azaltzen da. Oro har, antzekotasun semantikoen balioak nahiko orekatuta daude, hau da, balio guztietarako instantzia nahikoak daude. Hala ere, $[0,5]$ tarteko zenbaki naturaletan tontor batzuk aurkitzen dira 2.20 irudi-multzoko grafiko guztietan. Izan ere, antzekotasun semantikoak balio errealekin adierazten badira ere, balio horiek zenbaki arrunten bidez kalkulatu dira.

Demagun bost pertsonek bi esaldien arteko antzekotasun semantikoa estimatu behar dutela. Lau pertsonek 3 balioa estimatu dute, eta azkenekoak, berriz, 4 balioa. Bi esaldien antzekotasun semantikoa bost balioen arteko batez-bestekoa kalkulatzen da, 3.2 izanik. Kasu askotan bost pertsonek balio bera estimatzen dute, aipatutako tontorrak sortuz.

STS-B datu multzoan antzekotasun semantikoen balioak sortzeko estrategia bera jarraitu da, instantzia bakoitza bost pertsona ezberdinek aztertu dutelarik.



2.20 Irudia: *STS-B* datu multzoko instantzien antzekotasun semantikoen balioak. Instantzien balioak 6 azpimultzotan banatu dira: *Train-set*, *Dev-set*, *Test-set*, *News*, *Captions* eta *Forum*.



2.21 Irudia: Datu multzoa 9 zatitan banatu da, 2.3 taulako eskema jarraituz. Grafiko bakoitzean ardatz horizontalak instantzia bakoitzeko bi esaldiek konpartitzen dituzten hitzen proportzioa adierazten du. Ardatz bertikalak berdin jokatzen du, hitz hutsak kontuan hartzen direlarik bakarrik. Puntu bakoitzaren koloreak instantziaren antzekotasun semantikoa adierazten du: 0, 1, 2, 3, 4 eta 5.

Azkenik, 2.21 irudian antzekotasun semantikoak bi esaldiek konpartitzen dituzten hitzen proportzioekin duen zerikusia aztertzen da. Irudien goiburuko eta berrietan bi esaldiek partekatzen dituzten hitzen kopurua handitzen den heinean, antzekotasun semantikoa handitzen joan ohi da. Baina, hitz horiek gehienbat hitz hutsak direnean, antzekotasuna asko txikitzen da —konpartitzen dituzten hitzek ez dituztelako esanahi semantikorik—. Aurreko hau erraz ikusten da grafikoetan. Izan ere, konpartitzen dituzten hitz guztiak hitz hutsak direnean, hau da, grafiko bakoitzean ikus daitezkeen goi-ezker aldeko puntuek osatzen duten lerroetan, instantzia ia guztiek antzekotasun oso baxuak dituzte.

Foroetako instantzietan hau ez da beste azpimultzoetan bezain garbi azaltzen, ziurrenik hitz hutsen erabilera handiagoa delako.

Analisi hauek ez dira oso beharrezkoak egin diren esperimentuentzako, baina behintzat argi gelditzen da esaldien jatorriak hauen egitura baldintzatzen dutela; eta irudietako goiburukoetatik datozenak oinarrizko egitura bat jarraitzen dutela —*vSTS* ataza zerbait errazten delarik—.

Beste datu multzo batzuk

Datu multzo asko daude eskura ikertzaileek diseinatutako sistemek hizkuntzaren prozesamenduko hainbat ataza ikasi ahal izateko. Horietatik hainbat komentatuko dira, *STS* atazarekin erlazioa dutenak.

- *SNLI corpus*: datu multzo hau ingelesez idatzitako $5.7 \cdot 10^5$ esaldi-parez osatuta dago, *NLI* edo *Natural Language Inference* ataza ebazteko pentsatuta dagoena [4]. Ataza hau *STS*-ren antzekoa da; baina, antzekotasun semantikoak estimatu beharrean, bi esaldien arteko erlazioa aztertzen da.

  Erlazio horiek *entailment*, *contradiction* edo *neutro* izan daitezke —bi esaldien artean informazio osagarria lortu, kontradikzioak sortu, edota neutroak izanik, hurrenez hurren—. Beraz, *STS* ataza antzeko erregresio problema bat bada ere, *NLI* hiru klaseko sailkapen ataza bat da. Instantzia bateko klase bakoitza bost pertsonek eman duten iritziaren arabera definitu da, *STS-Benchmark* datu multzoan egin den moduan.

- *GLUE*: *General Language Understanding Evaluation* datu multzoak hainbat baliabide batzen ditu hizkuntzaren prozesamenduko sistemak entrenatu, ebaluatu eta aztertzeko [25]. Baliabide hauek bederatzi ataza ezberdinez osatuta daude, *STS* horietako bat izanik. Funtsean, sistema ezberdinen errendimendua aztertzeko sortua

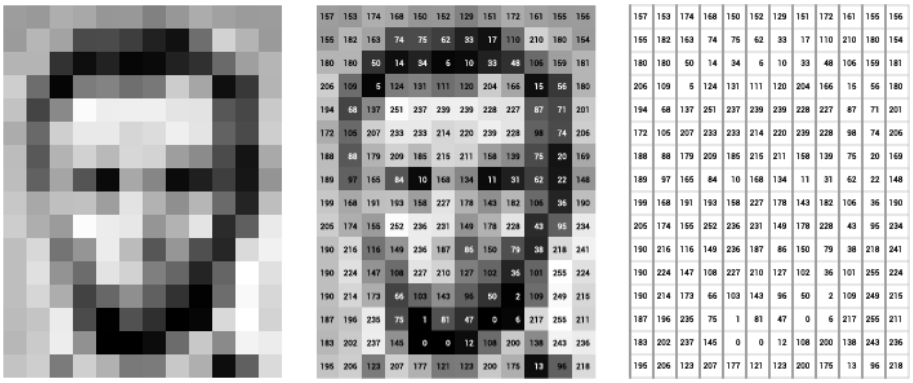
izan da, azterketa hori hizkuntza naturalek dituzten hainbat fenomeno linguistikoen gainean osatuz.

Gainera, sistema hauen errendimendua neurtzen duten sailkapen publiko bat jarri dute sarean, sailkapenean lehenengo postua edukitzeak hizkuntzaren prozesamenduko hainbat ataza ezberdinetan errendimendu handiko sistema bat izatea esan nahi du eta.

- *SuperGLUE*: *GLUE* datu multzoaren bertsio hobetu bat da, hizkuntzaren prozesamenduko ataza zailagoez osatuta, baliabide eguneratu eta saikapen berri batekin [26]. *BERT* bezalako sistemek lortzen dituzten emaitzak pertsonek ia lortzen dituztenak bezain onak direnez, *SuperGLUE* sortu behar izan dute gaur egungo sistemek hobekuntza tarte bat eduki ahal izateko.

2.2 Irudien errepresentazioak

Hitz-bektoreak ez bezala, irudiak egitura bi-dimentsionalak bezala tratatzen dira, hots, matrizeak, altuera eta zabalera jakin batzuk dituztenak eta pixel kopuruetan neurtzen direnak —2.22 irudia—.



2.22 Irudia: 12x16 pixeleko gris-eskala irudi bi-dimentsional bat. Pixel bakoitza 0 eta 255 arteko balio batekin esleitzen da, non kolore beltza 0 batekin eta kolore zuria 255 balioarekin definitzen diren.

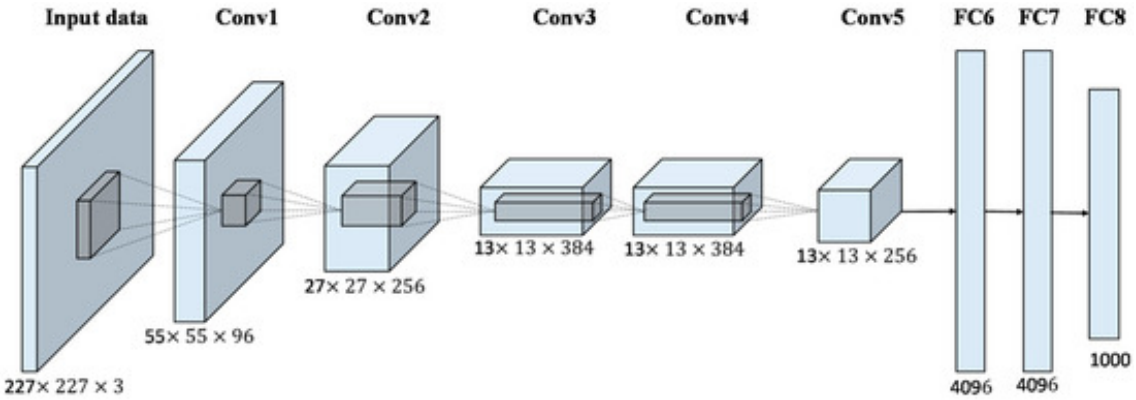
Proiektuan erabili diren irudiek *RGB* formatua erabiltzen dituzte. Formatu honetan pixel bakoitza 0 eta 255 arteko hiru balioz osatuta dago, pixel horiek dituzten gorri, berde eta urdin intentsitateak definitzen dituztelarik.

Irudi batean pixel baten kokapenak garrantzi handia du, pixel horien bizilagunen —edo hauen ondoko pixelen— artean erlazio edo ezaugarriak osatzen direlarik. Irudietan hain-

bat forma eraikitzeko lotura espazial hau garrantzitsua denez, horiek kontuan hartzen dituzten iragazki edo konboluzio deritzon eragiketak erabiltzen dituzten neurona-sareak erabili dira irudiak tratatzerako orduan.

2.2.1 Konboluzio-sareak

Azken hamarkada honetan irudiak tratatzerako orduan *konboluzio-sareak* izan dira gehien erabili diren neurona-sareak. Hauen erabilera 2012. urtean zabaltzen hasi zen, zortzi geruzako *AlexNet* konboluzio-sarearen arrakasta ikusiz [27]. 2.23 irudian sarearen eskema aurki daiteke. Sare hauen erabilera asko hazi da geroztik, neurona-sare mota ezagunenetarikoa bihurtuz.



2.23 Irudia: AlexNet neurona-sarearen eskema. Lehenengo bost geruzak konboluzio-geruzak dira —1, 2 eta 5 geruzek *max-pooling* eragiketa bat eginik—. Azkeneko hiruak, berriz, geruzadentsoak dira [27].

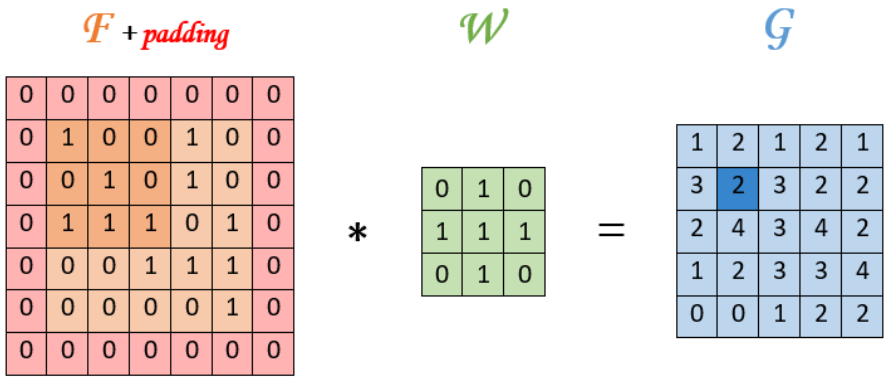
AlexNet irudien sailkatze problema baterako diseinatu zen, *ImageNet* datu multzoko irudiak sailkatzearena hain zuzen ere [28], 1000 klase ezberdin ezberdindu behar zituelarik. Horregatik, bere azkeneko geruzak mila elementu ditu, *softmax* funtzioa aplikatu ondoren klase bakoitza izateko probabilitateak lortzen baitira. Hortik aurrera sortu izan diren konboluzio-sare berriak *AlexNet*en egitura jarraitu dute, atazaren arabera hauen konplexutasun eta irteera-geruzak aldatuz —-erabili diren *VGG19* eta *Resnet152*, besteak beste—. Aipatu diren bi sistema hauek deskribatu baino lehen konboluzio eta *pooling* eragiketak azalduko dira, garrantzitsuak baitira sare hauetan.

Konboluzio bat geruza batean lokalki aplikatzen den eragiketa bat da, eragiketa hori behin eta berriz errepikatuz geruza horretan zehar. Funtsean, konboluzio batean bi matrizeen elementuen arteko biderketa bat egiten da, eta, ondoren, elementu guztiak batu egiten

dira, φ aktibazio-funtzio bat aplikatuz bukaeran —normalean *ReLU* dena—. Demagun $\mathbf{F}$ irudiari $\mathbf{W}$ iragazki edo konboluzio-matrize bat aplikatu nahi zaiola, $\mathbf{G}$ irudi bat lortzeko.

$$\mathbf{G}_{x,y} = \varphi(\mathbf{W} * \mathbf{F}_{x,y}) = \varphi(\sum_{s=-a}^{a} \sum_{t=-b}^{b} \mathbf{W}_{s,t} \cdot \mathbf{f}_{x-s,y-t}) \tag{2.21}$$

2.21 ekuazioko notazioari buruz, $\mathbf{F}_{x,y}$ sarrera irudiko x lerro eta y zutabeko pixelari deritzo. Erabiltzen diren iragazkien w zabalera eta h altuera bakoitiak direla kontuan izanik, $a = \lfloor \frac{w}{2} \rfloor$ eta $b = \lfloor \frac{h}{2} \rfloor$ bezala definitzen dira. Iragazkiaren aplikazioa 2.24 irudian azter daiteke.



2.24 Irudia: 3x3 tamainako iragazki baten aplikazioa 5x5 pixeleko irudi batean. Irteeran 5x5 pixeleko irudi bat lortzen da, sarrerako irudiari *zero-padding*ak gehitzen zaizkio eta —gorriz irudikatuta dauden lerro eta zutabeak—.

Konboluzio-geruza batean hainbat hiperparametro aukeratu behar dira, ondoren zerrendatzen direnak:

- *Kernel-size*: Erabiltzen den $\mathbf{W}$ iragazkiaren tamaina, normalean bakoitia dena. Iragazki karratuak erabiltzen direnez, $\mathbf{W}_{\text{size}} = w = h$ betetzen da. Irudiko adibidean $\mathbf{W}_{\text{size}} = 3$ betetzen da. Iragazkiaren tamaina eta kopuruek ikasi behar diren pisu kopurua baldintzatzen du, neurona-sare hauetan iragazkien balioak ikasten baitira.

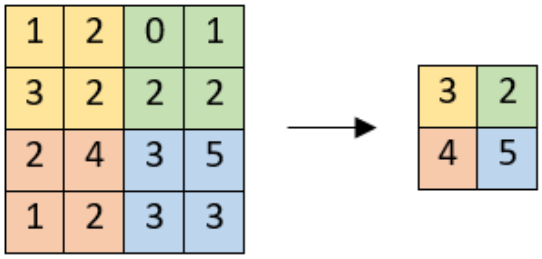
- *Filter-count*: $\mathbf{F}$ irudi edo geruzari aplikatzen zaizkion iragazki kopurua. Normalean, 2^x iragazki erabiltzen dira geruza bakoitzean, x edozein zenbaki arrunt izanik. Konboluzio-sareetako hasierako geruzetan iragazki gutxiago erabiltzen dira azkenekoetan baino. *AlexNet* sarean, adibidez, geruza bakoitzean 96, 256, 384, 384 eta 256 iragazki erabiltzen dira, lehenengotik azkenekora ordenatuta daudelarik.

- *Stride*: Aldagai honek iragazkien bi aplikazio ezberdinen artean zenbat pixeleko distantzia dagoen definitzen du. *Stride* handiagoa den heinean, $\mathbf{W}$ iragazkiak gutxiagotan aplikatu behar dira, eta, ondorioz, $\mathbf{G}$ irudia txikiago bihurtzen da. 2.24

irudian stride = 1 erabili da, 5x5 sarrera-matrize batetik 5x5 irteera-matrize bat lortuz. stride = 2 balitz, adibidez, irteera-matrizea 3x3 pixelekoa izango litzateke.

- *Padding*: Normalean konboluzioetan *zero-padding*a erabiltzen da **F** eta **G** irudien tamainak mantentzeko, stride = 1 izanik —2.24 irudian hau erraz ikusi daiteke— . *Zero-padding*ean zerokoak esleitzen zaizkie gehitutako lerro eta zutabeei, baina badaude beste *padding* mota batzuk, ertzeko balioak errepikatzen dituztenak adibidez. *Padding*ean gehitzen diren lerro eta zutabe kopurua iragazkien tamainaren araberakoa da.

Batzuetan konboluzio-geruza baten bukaeran *pooling* eragiketa bat burutzen da, lortutako irudien dimentsioa txikitzeko ahaleginean. Funtsean, *pooling* bat egitean pixel bizilagunak multzokatu eta balio batera konprimitzen dira. Balio horiek konprimitzeko hainbat eragiketa egin daitezke: handienarekin gelditu —2.25 irudia—, mediana kalkulatu etab.



2.25 Irudia: *Max-pooling* baten aplikazioa, non 2x2 tamainako pixel multzoak hartzen diren eta multzoko balio handienarekin gelditzen den. Kasu honetan stride = 2 erabili da.

Hala ere, konboluzio-geruza batean ez da irudi bakar bat tratatzen bakarrik. *Filter-count* aldagaia aipatu denean esan denez, 2.23 irudia berrikusten bada, hainbat irudiko pila bat landu behar dela ikusten da. Iragazki bakoitza aplikatzean aurreko geruzako irudi-pilaren irudi guztiak erabiltzen dira; beraz, errealitatean **W** iragazkiak hiru-dimentsionalak dira, hots, sakonera bat dute. Hau ere 2.23 irudian paralelepipedo txiki ilunago batzuekin irudikatzen da.

Konboluzio-sareak ikasi ondoren, geruza bateko iragazki bakoitza irudietako forma edo ezaugarri batzuen detekzioan kontzentratzen direla ikus daiteke. Sare hauetako geruzetan zehar aurrera joan ahala, ezaugarri hauek konplexuagoak bihurtzen dira —lerro edo kurba sinpleetatik hasiz, objektuak detektatzen joan arte—.

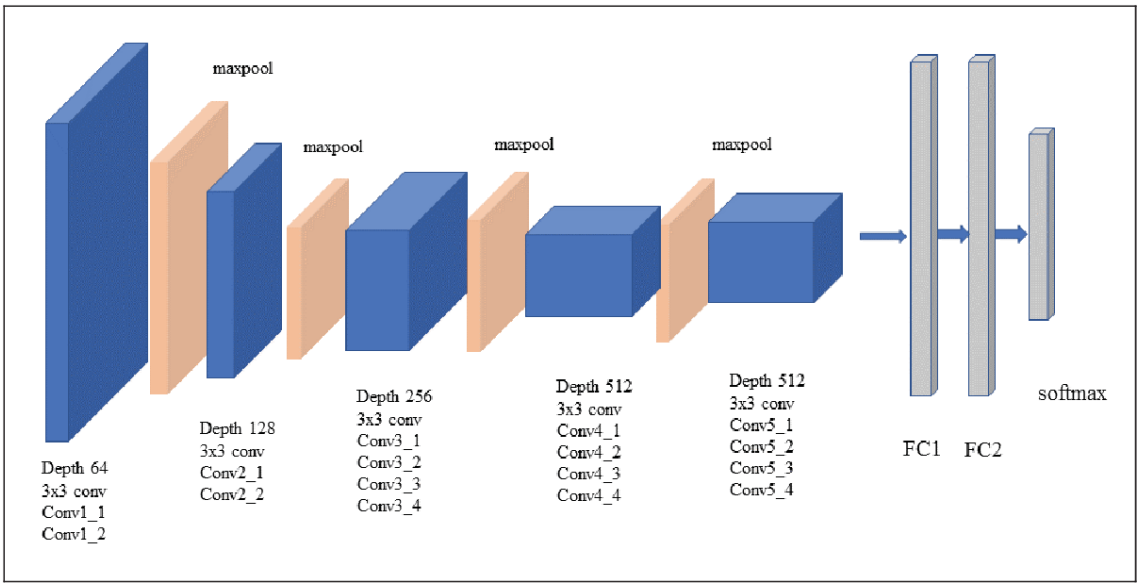
Konboluzio-sareek errendimendu altua dute irudiekin, baina hitz-bektore multzoekin ere errendimendu altua lortzen dute, esaldien sailkapenerako, adibidez [29]. Hala ere, proiektu honetan konboluzio-sareak irudien tratamenduan bakarrik aplikatu dira.

2.2.2 Erabilitako sistemak

Konboluzio-sareak *vSTS* atazarako erabili dira bakarrik, ondoren definituko diren *VSE++* sistemetan hain zuzen ere. Laburki, sistema hauetan erabili diren bi konboluzio-sare ezberdinak deskribatuko dira.

VGG19

VGG19 17 konboluzio-geruza eta 2 geruza-dentsoz osatuta dago [30]. Konboluzio-geruzak bost multzotan banatu daitezke, eta multzo bakoitzaren bukaeran *pooling* eragiketa bat burutzen da. 2.26 irudian 3x3 tamainako iragazkiak erabiltzen direla ikus daiteke, stride $= 1$ eta *zero-padding*a erabiliz. *Max-pooling* eragiketak, berriz, 2x2 pixel-multzoekin burutzen dira —azkenekoa izan ezik, 3x3 tamainakoa baita—, balio maximoarekin geldituz eta stride $= 2$ erabiliz.



2.26 Irudia: VGG19 sistemaren eskema, [31] web orritik eskuratua.

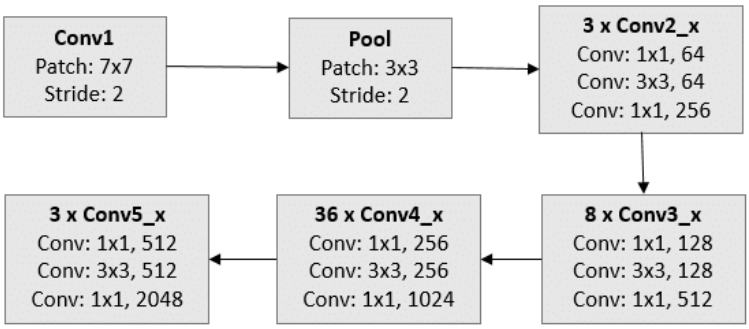
Azkenko *pooling* eragiketa egin ondoren, bi geruza-dentso aurkitzen dira. Lehenengoak 4096 elementuko irteera du; eta bigarrenak, berriz, sailkatu nahi diren klaseen elementu kopurua, —azkeneko hau atazaren menpekoa delarik, eta *AlexNet* sarearen kasuan 1000 dena—. Azaldu diren iragazkien ezaugarriak 48x48 pixeleko sarrerako irudientzako pentsatuta dago, baina hauek aldatu egin daitezke tamaina hori igo nahi bada, ikasi behar diren parametro kopurua handituz noski. 2.26 irudiko eskeman, adibidez, 43 milioi parametro inguru ikasi behar dira.

ResNet152

ResNet152 konboluzio-sareak aurrekoak baino errendimendu zerbait altuagoa lortzen du, konboluzio-geruza kopurua asko handituz —sareak, guztira, 152 geruza baititu—. *VGG19* sareak bezalaxe, aldaera asko ditu; baina hainbat lanetan ikus daitekeenez, erabili den egiturak errendimendu hoberenetarikoa ematen duela ikusi da [32] —geroz eta geruza gehiago, orduan eta emaitza hobeak, baina sistemaren konplexutasun handiagoa—. Oro har, *VGG* motako sistemek baino emaitza hobeagoak ematen dituzte *ResNet* sareek.

Horrexegatik, *VSE++* sistemaren egileek egitura hau erabili zuten bere esperimentuetarako [33]; eta lan honetan egile horien aukera erabili da.

Funtsean, *Residual Network* batean geruza bateko eragiketak egin ondoren, geruza horren sarrera eta irteera datuak batzen dira, datu horiek dimentsionalitate bera dutela kontuan izanik. Modu sinple batean esanda, geruza batek f funtzio bat burutzen badu, geruza horren irteeran $f(x)+x$ egiten da, x sarrerako datua izanik. Aurrez aldetik azaldutako *Transformer*rek modu berean jokatzen dute, 2.13 irudian azaldu den bezala.



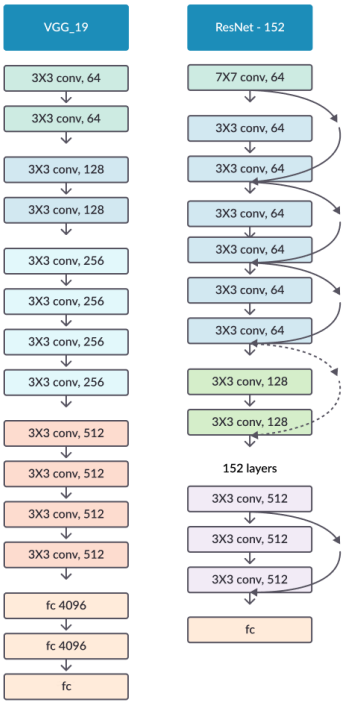
2.27 Irudia: *ResNet152* sistemaren eskema, non konboluzio-geruzak bost multzotan banatzen diren. Lehenengo konboluzioak 7x7 tamainako iragazki bat erabiltzen du, stride $= 2$ izanik. Ondoren, *max-pooling* bat burutzen da, 3x3 pixel-multzoekin eta *stride* berarekin. Ondoren hainbat konboluzio-geruza multzo ezberdin errepikatzen dira, 3, 8, 36 eta 3 aldiz, hauen iragazki kopurua multzo bakoitzean handituz —[64, 64, 256] iragazki kopuruko konboluzio-geruza multzotik [512, 512, 2048] iragazki kopuruko batera— [32].

ResNet152 sistemaren kasu gehienetan —lehenengoan izan ezik—, hiru konbuluzio-geruza burutzen dira sarrera eta irteerako datuak batu baino lehen, hots, f hiru konboluzio-geruzek osatzen duten funtzioa da. Hauxe bera 2.27 irudian irudikatzen da, non konboluzio-geruza bakoitzaren iragazki tamaina eta hauen kopurua erakusten den.

Noizean behin, konboluzio-geruza multzo artean, batez-besteko *pooling*a aplikatzen da, pixel-multzoetako batez-bestekoa kalkulatuz eta irudiaren tamaina txikituz.

Azkenik, geruza-dentso batekin sistemaren irteera-datuak lortzen dira, atazaren arabera aldatzen dena. Deskribatu den egiturarekin, 112x112 tamainako sarrerako irudiak ematen zaizkio sareari, *VGG19* sistemarena baino tamaina handiagoa dena.

Bien arteko konparazio grafiko batekin amaitzeko, 2.28 irudia ikus daiteke.



2.28 Irudia: *VGG19* eta *ResNet152* sistemen konparaketa grafikoa, [34] artikulutik eskuratua.

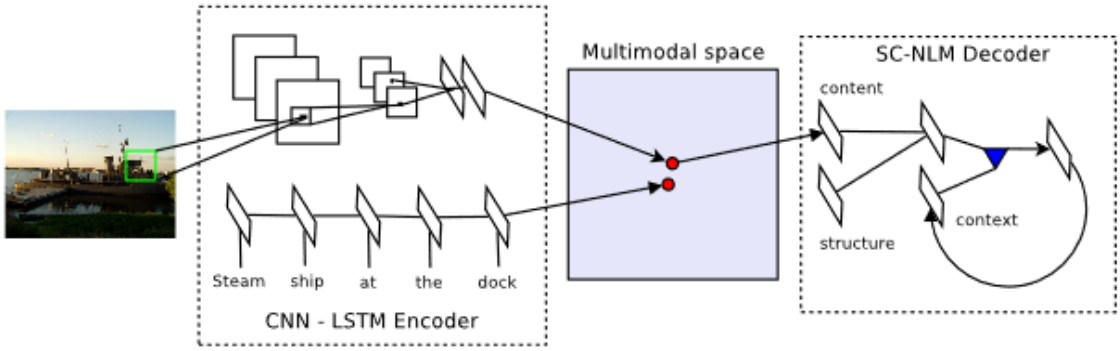
2.3 Modalitate anitzeko errepresentazioak

Pertsonen arteko komunikazioan informazioa ez da inoiz osoa, pertsonek aurretik duten ezagutzarekin osatu eta interpretatzen dira eta. Makinek pertsonekin komunikatu ahal izateko ezinbestekoa da ezagutza guzti hori bereganatzea; baina, testua ez dago errealitateari lotuta. Berriki, gabezia horiek irudi eta bideo multzo handietatik ezagutza induzituz osatu nahi izan da, irudietan azaltzen den ezagutza bisuala erabiliz, eta testuen errepresentazioa mundu fisikoari lotuz irudien bidez, *grounding* deritzon prozesuan.

Grounding prozesu hori burutzeko modu bat modalitate anitzeko errepresentazioak erabiltzea da. Dokumentu honetan aipatu den bezala, modalitate anitzeko errepresentazioek bi modalitate edo gehiagoko adierazpenak espazio berean kokatzea ahalbidetzen dute. Kasu honetan errepresentazio horiek testu eta irudien modalitateak batzen dituzte.

2.3.1 *VSE++*

Visual-Semantic Embedding++ sistemak irudi eta testuen errepresentazioak espazio berean sortzen ditu [33], *VSE0* arkitekturaren hobekuntza bat delarik [35]. Biek irudi eta goiburukoen berreskurapena edo *image and caption retrieval* atazak burutzen dituzte. Irudien berreskurapenaren kasuan goiburuko bati dagokion irudi egokiena estimatu behar da —eskura dauden irudi multzoetatik hautatu behar dena—, eta goiburukoen berreskurapenean, aldiz, irudi bati dagokion goiburuko egokiena estimatu behar da. *VSE++* bakarrik erabili bada ere, lehenik eta behin *VSE0* azalduko da; eta, ondoren, *VSE++* sistemarekiko ezberdintasun nagusiak aipatuko dira.



2.29 Irudia: *VSE0* sistemaren eskema, kodetzaile eta deskodetzaileetan zatitua [35]. Kodetzailearen atalean, irudiaren errepresentazioa lortzeko *CNN* bat erabiltzen da, eta esaldiena lortzeko, berriz, *LSTM* gelxakak erabiltzen dituen neurona-sare errepikakor bat.

VSE familiako sistemetan irudiak eta testuak banatuta prozesatzen dira hasiera batean, bakoitza bere aldetik. Behin bien errepresentazioak lortzen direnean errepresentazio horiek transformatu egiten dira modalitate anitzeko errepresentazioen espazio batera, matrizeen arteko biderketen bidez —2.29 irudia—. Hortaz, irudi eta goiburukoak espazio berean definituta daudenez, **c** goiburuko baten irudirik semantikoki antzekoena bi errepresentazioen arteko kosinuaren antzekotasun handiena duena aukeratuz eskuratu daiteke, hau da, espazioan gertuen dagoen irudiaren errepresentazioa hautatuz.

Ondoren, hizkuntza-eredu bat ikasten duen neurona-sarea gehitzen zaio; baina, proiektuaren esperimentuetako ez da garrantzitsua, atal hau *VSE* arkitekturaren deskodetzailea baita.

Beraz, proiektu honetan *VSE* sistemaren kodetzailea erabiliko da bakarrik, kodetzaile honen irteeran lortzen diren modalitate anitzeko errepresentazioak nahi dira eta. Kodetzaile hau entrenatzeko orduan *image-caption retrieval* atazak erabili dira; non, datu multzo ja-

kin bateko irudi edo testu bat emanik, dagokion testu edo irudi parea zein den aukeratu behar den —*caption-retrieval* edo *image-retrieval*, hurrenez hurren—.

Ataza honetarako *Hinge* galera-funtzio bat diseinatu daiteke modalitate anitzeko espazioa eraikitzeko. Demagun N irudi eta testu pareko azpimultzo edo *batch* bat eskura dagoela, azpimultzoko i. parea $\mathbf{i}_i$ irudia eta $\mathbf{c}_i$ goiburukoa izanik. *VSE0* sisteman 2.22 eta 2.23 ekuazioek galera-funtzio hori definitzen dute, *Hingen-batura* —*Sum-of-Hinges* edo *SH*— deritzona.

$$\text{loss} = \frac{1}{N}\sum_{n=1}^{N} l\left(\mathbf{i}_n, \mathbf{c}_n\right) \tag{2.22}$$

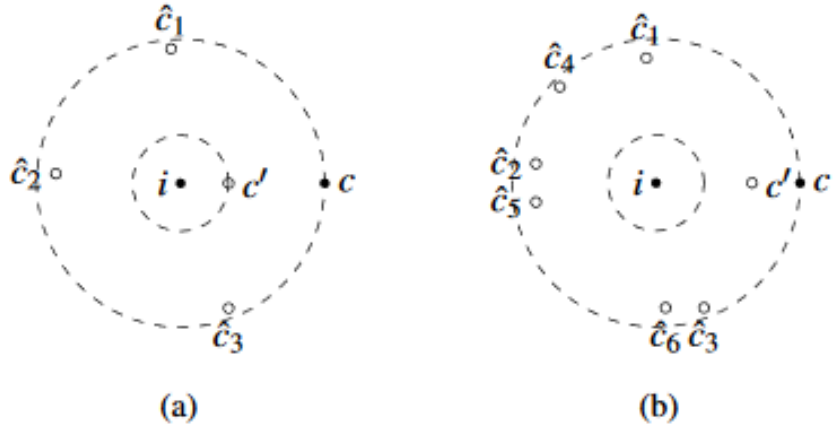
$$l_{\text{SH}}\left(\mathbf{i},\mathbf{c}\right) = \sum_{\hat{\mathbf{c}}}\left[\alpha - s(\mathbf{i},\mathbf{c}) + s(\mathbf{i},\hat{\mathbf{c}})\right]_+ + \sum_{\hat{\mathbf{i}}}\left[\alpha - s(\mathbf{i},\mathbf{c}) + s(\hat{\mathbf{i}},\mathbf{c})\right]_+ \tag{2.23}$$

Hingen-batura honetan α konstante positibo txiki bat da, 2.23 galera-funtzioari errore-marjina bat gehitzen diona. $s(\mathbf{i},\mathbf{c})$ funtzioak irudi eta goiburuko baten errepresentazioen arteko kosinuaren-antzekotasuna adierazten du. $\hat{\mathbf{i}}$ eta $\hat{\mathbf{c}}$, berriz, bere goiburuko edo irudien pareak ez diren *batch*eko pare guztiek osatzen dute —kasu negatiboek—, hurrenez hurren. Azkenik, $[x]_+$ eragiketa $\max\{0,x\}$ funtzioaren baliokidea, hau da, *ReLU* funtzioa da.

Funtsean, 2.23 ekuazioa bi zatitan banatzen da. Lehenengoan goiburu berreskurapena burutzen da, non galera-funtzioak itzultzen duen balioa handiagoa izango den $\mathbf{i}$ irudiaren $\mathbf{c}$ goiburuko positiboa baino gertuago dauden $\hat{\mathbf{c}}$ goiburuko negatibo bakoitzeko. Horrela, eredua entrenatzerako orduan balio hori txikitzen joango da pare bakoitzaren arteko errepresentazioak gerturatzen doazen heinean. Bigarrenean, berriz, irudi-berreskurapena burutzen da, irudien pare egokiak aztertu beharrean aurreko modu berean goiburukoenak aztertzen dituelarik.

SH galera-funtzioak ematen duen errendimendua ona bada ere, 2.30 irudian adierazi bezala, adibidez, espero ez diren emaitzak lortzen dira. Hori dela eta, *Hingen-maximoa* —*Max-of-Hinges* edo *MH*— izeneko galera-funtzioa definitu da *VSE++* sistemarako, kasu horietako emaitzak zuzentzeko asmoarekin.

$$l_{\text{MH}}\left(\mathbf{i},\mathbf{c}\right) = \max_{\mathbf{c}'}\left[\alpha - s(\mathbf{i},\mathbf{c}) + s(\mathbf{i},\mathbf{c}')\right]_+ + \max_{\mathbf{i}'}\left[\alpha - s(\mathbf{i},\mathbf{c}) + s(\mathbf{i}',\mathbf{c})\right]_+ \tag{2.24}$$



2.30 Irudia: Irudi baten errepresentazioaren gertueneko goiburukoak. Irudi horren pare positiboa edo **c** beltzez betetako zirkulu batez adierazten da, besteak goiburuko negatiboak edo $\hat{\mathbf{c}}$ izanik. Goiburukorik negatiboena edo $\mathbf{c}'$ gertuago dago *(a)* kasuan bestean baino. Hala ere, *SH* galera-funtzioa handiagoa da *(b)* kasuan, pare positiboa baino gertuago goiburuko negatibo asko daudelako. Hori *MH* galera-funtzioak konpontzen du, kasu negatiboenari erreparatzen dio eta bakarrik. Irudia [33] artikulutik hartu da.

Max-of-Hinges galera-funtzioa 2.24 ekuazioan definitzen da, non $\mathbf{i}'$ eta $\mathbf{c}'$ gertuen dauden kasu negatiboak edo kasu-negatiboenak diren, 2.25 ekuazioetan definituta daudelarik.

$$\mathbf{i}' = \arg\max_{\mathbf{j}\neq\mathbf{i}} s(\mathbf{j},\mathbf{c}) \tag{2.25a}$$

$$\mathbf{c}' = \arg\max_{\mathbf{d}\neq\mathbf{c}} s(\mathbf{i},\mathbf{d}) \tag{2.25b}$$

Galera-funtzioak alde batera utziz, *VSE0* eta *VSE++* sistemen arteko beste aldaketa nagusia esaldiak kodetzeko *GRU* gelaxkak erabiltzen duen *RNN* sarea hautatu dela da, *LSTM*ak erabili beharrean.

Oro har, *VSE++* sistemarekin lortzen diren emaitzak hobeagoak direnez, lortzen diren modalitate anitzeko errepresentazio pareen arteko antzekotasun semantikoak handiagoak izango direla aurreikusten da; eta, hortaz, sistema berrienaren errepresentazioak bakarrik erabili dira proiektuko esperimentuetan.

2.3.2 Modalitate anitzeko datu multzoak

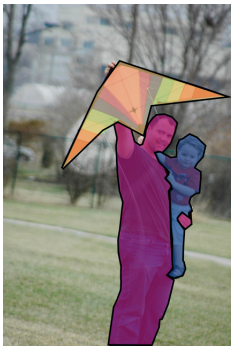
Modalitate anitzeko datu multzoak ikasteko irudi eta goiburuko pareez osatutako datu multzo erraldoiak behar dira. Hurrengo lerroetan bi adibide agertzen dira: *MS-Coco* eta

Flickr30k. Horietatik, sortu den arkitektura berria entrenatzeko lehenengoa aukeratu da.

MS-Coco

Printzipioz, *MS-Coco* datu multzoa irudietan *objektuen antzematea* burutzeko pentsatua badago ere [36], modalitate anitzeko errepresentazioak ikasteko erabili daiteke, datu multzoko irudi bakoitza bost goiburuko ezberdinekin dator eta [37].

Gaur egun, *Microsoft*en datu multzo hau 328124 irudiz osatuta dago, hiru azpimultzotan banatua: 165482 irudiko *train-set*, 81208-ko *development-set* eta 81434-ko *test-set*. Azpimultzo guztietako irudi askotan detektatzeko hainbat objektu edo klase dituzte, irudien goiburukoei konplexutasuna emanez —ez dira irudi sinpleak eta—.



1. a boy and a man holding a kite outside.
2. a man holding a toddler has a rainbow kite.
3. a father and son fly a rainbow kite.
4. a man and child prepare to fly a kite.
5. a man flying a kite and holding a small boy.

2.7 Taula: MS-Coco datu multzoko instantzia: irudi originala, definituta dauden klaseen segmentazioak —pertsonak eta kometa— eta datozen bost goiburukoez osatua —letra xehez idatzita daudenak—.

Datu multzoko instantzia baten edukia 2.7 taulan aurki daiteke. Irudien segmentazio eta goiburukoak eskuz definitu dira, eta, oraingoz, 91 klase ezberdin aurki daitezke segmentazio horietan.

Flickr30k

Flickr30k irudi eta hauen goiburukoez osatutako modalitate anitzeko datu multzoa da [38], aurretik sortua zegoen *Flickr8k* datu multzoaren hedapena delarik [39]. *Flickr30k* datu multzoak 158915 goiburuko ditu, 31783 irudi deskribatzen dituztenak, irudi bakoitzak bost goiburuko dituelarik.

Bi datu multzoetako irudietan pertsonek eguneroko bizitzan izaten dituzten aktibitateak aurkitzen dira, eta hauen goiburuko bakoitzean irudiko hainbat ezaugarri eta hitz ezberdinak erabiltzen dira.

Flickr30k datu multzoa *MS-Coco* baino nahiko txikiagoa da, ia hamar aldiz txikiagoa instantzia kopuru aldetik. Gainera, *MS-Coco*k barietate handiagoa du bere irudietan, irudi guztietan ez baitira hainbat ekintza egiten dituzten pertsonak bakarrik ikusten eta.

Barietate hori dela eta, *MS-Coco* arkitektura berria entrenatzeko egokiena dela suposatu da.

2.4 *Generative Adversarial Networks*

Sare sortzaile antagonikoa —Generative Adversarial Network edo *GAN*— bi neurona-sare sakonez osatutako arkitektura da [40], bi sareek aurkako helburuak dituztelarik, hortik datorrelarik antagoniko hitza. Sare hauen potentziala oso handia da, datu multzo batetik hasita datu horien distribuzioa ikasteko ahalmena dute eta —distribuzio bereko irudiak sortuz, adibidez—.

GAN hauen ideia nagusia bi neurona-sare bata bestearen aurka jartzea da. Lehenengoak, datu multzoaren p_{data} distribuzioa jarraituz instantzia berri bat sortuko du, *GAN* arkitekturaren *Generator* edo sortzailea izanik. Bigarrenak, berriz, sortzaileak sortzen dituen instantzia berriak orijinaletatik desberdintzen ikasi behar ditu, arkitekturaren *Discriminator* edo diskriminatzailea bihurtuz.

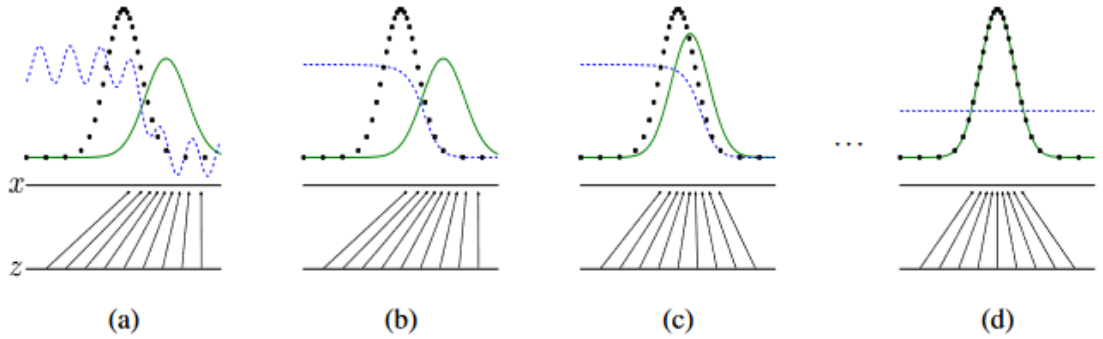
G sare sortzailearen irteera ikasi duen distribuzioaren tamainakoa da. 227x227 pixeleko *RGB* irudiez osatuta badago distribuzio hori, tamaina horretako irudi bat sortuko du. D diskriminatzaileak, berriz, probabilitate bat bakarrik ematen du, $[0,1]$ artekoa. 0 bat D sarearen sarrerako instantzia G sareak sortu duelaren estimazioa da; eta batekoa sarrerako instantzia jatorrizko datu multzotik datorrelarena.

Hasiera batean bi sare hauek ausazko emaitzak ematen badituzte ere, galera-funtzio jakin batzuen bitartez bi sareek bete behar dituzten zereginak ikasten hasiko dira. Ikasketa-prozesuan G sarearen helburu nagusia D sarearen huts egiteak maximizatzea da, hau da, $\log(1-D(G(\mathbf{z})))$ minimizatzea $\mathbf{z}$ sortzailearen sarrerako *zarata-bektorea* izanik. D sarearen helburua, berriz, estimazio zuzenak burutzeko probabilitatea maximizatzea da. Hortaz, bi jokalariko *mini-max* algoritmo bat jarraitzen da, sare bakoitza $V(G,D)$ funtzioa minimizatu eta maximizatzen saiatzen delarik —2.26 ekuazioa—.

$$\min_G \max_D V(G,D) = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))] \tag{2.26}$$

2.26 ekuazioko notazioari begira, $\mathbf{x}$ erabiltzen ari den datu multzoaren instantzia bat da, eta ekuazioaren $\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}[\log D(\mathbf{x})]$ atalean $D(\mathbf{x})$ kalkulatzen da, non D sarearen $\mathbf{x}$ sarrerako datuak datu multzoaren instantzien distribuzioa jarraitzen duela estimatzen den, edo behintzat horien antzekoa dela.

Ikasketa-prozesuan zehar G eta D sareek sortu eta ontzat estimatzen dituzten instantzien distribuzioak aldatzen doaz. Aldaketa hori 2.31 irudian ikusi daiteke.



2.31 Irudia: Lerro horizontal baxuenak $\mathbf{z}$ zarata-bektoreak lortzen diren domeinua adierazten du, kasu honetan uniformeki aukeratzen direnak. Domeinu horretatik ateratzen diren geziak $\mathbf{x} = G(\mathbf{z})$ mapaketa adierazten dute, irteerako $\mathbf{x}$ balioak G sortzailearen p_g distribuzioaren dentsitate handiko eskualdetan kontzentratzen direlarik —distribuzio hori lerro berdean adierazita dago—. Datu multzoak jarraitzen duen p_{data} distribuzioa puntu beltzez irudikatzen da, eta G sare sortzaileak distribuzio hori kopiatzeko helburua du, irudietan zehar prozesu hori ikusten delarik. Azkenik, diskriminatzailearen estimazioak marratxo urdinez adierazten da, ikasketa-prozesuan zehar egonkortzen doalarik. Izan ere, $p_g = p_{\text{data}}$ betetzean diskriminatzaileak estimazio erdiak gaizki egingo ditu batez-beste, instantzia guztiak distribuzio berekoak izango baitira [40].

Azkeneko irudian ikasketa-prozesua ondo burutzen deneko kasua irudikatzen da bakarrik, bi sareak abiadura berean ikasten dutelarik. Hau ez da zergatik gertatu behar; izan ere, hasiera batean diskriminatzaileak errazago detektatzen ditu instantzia faltsuak, errealekiko oso ezberdinak baitira. Horrek diskriminatzaileari abantaila gehiegi ematen dio, askotan sortzailea p_{data} distribuzioa ikasi ezinik gelditzen baita. Beraz, ikasketa-prozesuan bi sareak etengabe ikasten doazela ziurtatu behar da. Hau, askotan, sortzailearen pisuak maizago eguneratzen lortzen da, hots, G sarearen atzeranzko propagazioa D-rena baino maizago burutuz.

[40] artikuluan azaldu zirenetik, *GAN*ak zerbait aldatu dira. Lehenengo artikulu horretan sortzaile eta diskriminatzaileak geruza anitzeko pertzeptroiak baziren ere, gaur egun

konboluzio-sareak aurki daitezke sare sortzaile bezala. Gainera, gaur egun sortzaile baten sarrera zarata-bektore bat izan beharrean irudiak erabiltzea arrunta bihurtu da, ondoren azalduko den *DiscoGAN* arkitekturan bezala.

2.4.1 *DiscoGAN*

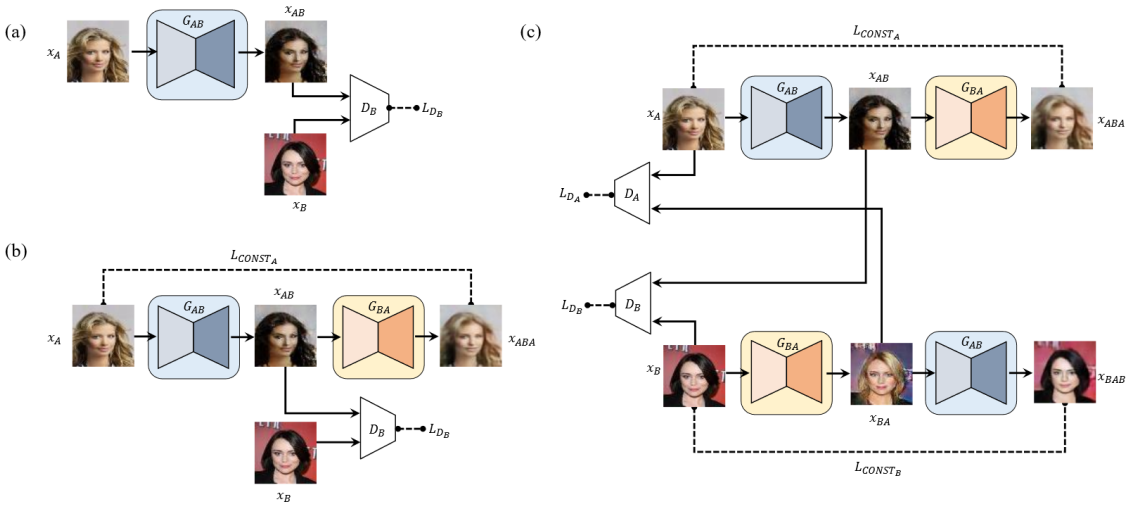
GAN sareetan oinarrituz, *DiscoGAN* arkitektura bi domeinu ezberdinen arteko mapaketa bijektiboa eraikitzeko diseinatu da [41]. Funtsean, domeinu ezberdineko bi datu multzo izanik, bien arteko transformazioak burutu ditzake, hau da, domeinu bateko irudi bat beste domeinura eraman dezake, 2.32 irudiko adibidean bezalaxe. Sistema honek bi domeinuen instantziez osatutako datu multzoekin funtzionatzen du —instantzia lerrokaturik behar gabe—, hots, irudiak ez daude parekatuta, ataza ikasteko beharrezkoak diren datu multzoak eraikitzea sinpleagoa bihurtuz. Oraingoz, *DiscoGAN* arkitektura irudietara bakarrik aplikatu da.

2.32 Irudia: Adibide hauetan poltsa eta zapatilen arteko transformazioak ikus daitezke. Transformazioak burutzean sarrerako irudien hainbat ezaugarri mantentzen direla ikus daiteke —koloreak eta estiloak, adibidez— [41].

Demagun bi funtzio ikasi nahi direla, G_{AB} eta G_{BA} deituko direnak. G_{AB} funtzioak A domeinu jakin bateko instantzia bat B domeinura transformatzen du. G_{BA} funtzioak, berriz, alderantziz funtzionatzen du, Btik A domeinura transformatuz. Funtzio horiek bi sortzaile ezberdinek ebatziko dituzte, izen bera edukiko dutenak. $\mathbf{x}_A$ A domeinuko irudi bat izanik, B domeinura mapaketa egitean $\mathbf{x}_{AB} = G_{AB}(\mathbf{x}_A)$ irudia lortzen da. Irudi berri hori A domeinura berriro eramateko $\mathbf{x}_{ABA} = G_{BA}(\mathbf{x}_{AB})$ eragiketa burutu behar da. Gainera, $G_{BA} \circ G_{AB}(\mathbf{x}_A) = \mathbf{x}_{ABA} = \mathbf{x}_A$ propietatea mantendu nahi da, hots, funtzio bat bestearen alderantzizkoa izatea. Pareko pauso bat burutu daiteke $\mathbf{x}_B$ irudiarekin, $\mathbf{x}_{BA}$ eta $\mathbf{x}_{BAB}$ lortzeko.

Ondoren, oinarrizko *GAN*etan bezala bi diskriminatzaile erabiltzen dira —bat domeinu bakoitzeko—, sortutako irudiak orijinalekin ezberdintzeko, D_A eta D_B deiturikoak. Eragiketa guztiak grafikoki 2.33 irudian erakusten dira; ondorengo galera-funtzioak erabiliz:

- L_{CONST_A} eta L_{CONST_B} berreraikitze galera-funtzioak, bi irudien arteko ezberdintasunak minimizatzen dituztenak, 2.27 ekuazioetan azaltzen direnak:



2.33 Irudia: (a) Oinarrizko *GAN*, aurreko azpiatalean deskribatu dena. (b) *GAN* berreraikitze galera-funtziaroekin. (c) Atal honetako *DiscoGAN* arkitektura [41].

$$L_{\text{CONST}_A} = d(G_{BA} \circ G_{AB}(\mathbf{x}_A), \mathbf{x}_A) \tag{2.27a}$$

$$L_{\text{CONST}_B} = d(G_{AB} \circ G_{BA}(\mathbf{x}_B), \mathbf{x}_B) \tag{2.27b}$$

- L_{G_A} eta L_{G_B} sortzaileen galera-funtzioak, sortutako irudien mapaketa zuzentzeko definitzen dena, 2.28 ekuazioetan azaltzen direnak:

$$L_{\text{G}_A} = -\mathbb{E}_{\mathbf{x}_B \sim P_B}[\log D_A(G_{BA}(\mathbf{x}_B))] \tag{2.28a}$$

$$L_{\text{G}_B} = -\mathbb{E}_{\mathbf{x}_A \sim P_A}[\log D_B(G_{AB}(\mathbf{x}_A))] \tag{2.28b}$$

- L_{D_A} eta L_{D_B} diskriminatzaileen galera-funtzioak, sortutako irudiak instantzia originaletatik ezberdintzeko erabilia, 2.29 ekuazioan definitua daudenak:

$$L_{\text{D}_A} = -\mathbb{E}_{\mathbf{x}_A \sim P_A}[\log D_A(\mathbf{x}_A)] - \mathbb{E}_{\mathbf{x}_B \sim P_B}[\log(1 - D_A(G_{BA}(\mathbf{x}_B)))] \tag{2.29a}$$

$$L_{\text{D}_B} = -\mathbb{E}_{\mathbf{x}_B \sim P_B}[\log D_B(\mathbf{x}_B)] - \mathbb{E}_{\mathbf{x}_A \sim P_A}[\log(1 - D_B(G_{AB}(\mathbf{x}_A)))] \tag{2.29b}$$

Aurreko ataleko oinarrizko *GAN*etik hasita, *DiscoGAN* deskribatzen den [41] artikuluan arkitektura zerbait aldatu da bi domeinuko mapaketa egin ahal izateko. Lehen esan bezala, sortzaileen sarrerako zarata-bektoreak erabili beharrean domeinu jakin bateko irudiak

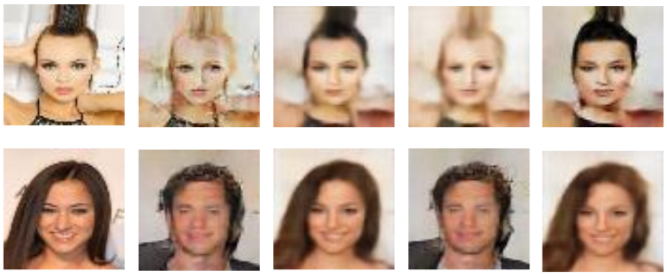
erabili dira. Gainera, bi zentzuzko mapaketa egiteko bigarren sortzaile bat ipini da, berreraikitze galera-funtzio bat gehituz. Honela, 2.33 irudiko (b) arkitektura lortzen da.

Hasiera batean arkitektura hori bi noranzkoko mapaketa ikasteko nahikoa zela pentsatu bazen ere, sistema honek ez du mapaketa bijektiboa bermatzen, ez baitu $G_{BA} \circ G_{AB}(\mathbf{x}_A) = \mathbf{x}_{ABA} = \mathbf{x}_A$ kondizioa zorrozki betetzen.

Hori konpontzeko *DiscoGAN* arkitektura diseinatu dute. Izan ere, bi domeinuetan berreraikitze galera-funtzioak erabiltzeak sare sortzaileek bere sarrera eta irteeretako instantzien artean ezaugarriak mantentzea ekartzen du. Kontuan izan behar da, 2.33 irudian lau sortzaile daudela ematen badu ere, bi sortzaile ezberdin erabiltzen direla bakarrik —G_{AB} eta G_{AB}—, bi sortzaile pare horiek pisuak partekatzen dituztelarik.

Kasu honetan erabili diren sortzaile eta diskriminatzaile guztiak konboluzio-sareak dira. *DiscoGAN* arkitekturan 64x64 tamainako irudiak erabili direla aipatzea komeni da, nahiko txikiak baitira. Hala ere, geruzetan aldaketa txiki batzuk eginik eta ikasketa-prozesu luzeago batekin irudi handiagoak erabili ahal dira.

Azkenik, *DiscoGAN*ek domeinuko distribuzioen mapaketa egiten duenez, mapaketa burutzean ezaugarri batzuk mantentzen baditu ere, beste domeinuko hainbat ezaugarri arrunt aplikatzen dizkie irudiei. Beraz, eragiketa hauek behin eta berriro aplikatzean irudi askotan ikusten ez diren eta hasierako irudian agertzen diren ezaugarriak galtzen joaten dira —2.34 irudia—.



2.34 Irudia: $\mathbf{x}_A \rightarrow \mathbf{x}_{ABABA}$ transizioaren bi adibide —prozesuan irudi horiek degradatzen doazelarik, batez ere irudiaren atzealdea—. Lehenengo lerroan domeinu aldaketa ilearen kolorea da. Bigarrenean, ordea, generoa da aldatzen dena [41].

3. KAPITULUA

vSTS datu multzoa

NLU edo giza-hizkuntzen ulermen automatikoa lortu ahal izateko betebehar nagusia testuen esanahiak mundu fisikoarekin lotzea da, testuan dagoen informazio inplizitua osatzeko ahaleginean [42]. Mundu fisikoarekin lotura ezartzeko irudi eta testu pareak dituzten datu multzoak erabili daitezke. Orain dela urte gutxi batzuk arte hizkuntza eta *konputagailu bidezko ikusmena* lantzen dituzten atazak ez ziren estentsiboki lantzen. Hala ere, neurona-sare sakon, konputagailu bidezko ikusmena eta hizkuntzaren prozesamenduaren aurrerapenek ikertzaileek alor honetan duten interesa piztu dute. Hainbat ataza eskura izanda, *vSTS* ataza aukeratzea hurrengo puntuek motibatu dute:

- **Testuen ulermena**: Hitz-bektoreen arrakastak testu luzeagoak modu berean errepresentatzeko metodoak sortzearen nahia ekarri du. *vSTS* atazan testuen arteko antzekotasun semantikoa aztertzen da, dagokien irudiekin lagunduta burutzen dena. Hortaz, lortzen diren errepresentazioen esanahia aztertzeko egokia bihurtzen da.

- **Modalitate aniztasuna**: Aurreko puntuarekin batera, azken urteetan testuen ulermena garatzeko hainbat ataza ezberdin definitu dira —*vSTS*, besteak beste—, horietako hainbat irudiez baliatzen direlarik. Horrela, testuen ulermena ikusmenaren esparruarekin lotzen da, hurrengo puntua sendotuz.

- **Osagarritasuna**: Modalitate anitzeko errepresentazioak erabiltzea bai testu eta baita irudien errepresentazioak ere hornitu ditzake. Goiburuko batek ekar dezaken informazioa bere irudiarekin osatu daiteke, bi modalitateko errepresentazioak osagarriak direlarik beraien artean. Osagarritasun hau *language grounding* problemaren ebazpenaren gakoa izan daiteke.

vSTS ebatzi ahal izateko erabili diren sistemak ataza horretan entrenatu behar dira, eta ikasketa-prozesu hori burutzeko datu multzo bat behar da. Ondorengo atalean *IXA* ikerketa-taldean eraiki den *vSTS* datu multzoa nola sortu eta hedatu den azaltzen da. Proiektu honetan datu multzoaren sorreran edo hedapenean modu zuzen batean ez bada parte hartu ere, datu multzoa nola eta zein irizpide erabiliz eraiki den aztertu da. Gauzak sinplifikatzearren [5] artikuluan aipatzen den datu multzoari *vSTS v1.0* deituko zaio; eta hedapena burutu ondoren eraiki zen datu multzoari *vSTS v2.0* —azkeneko hedapen hau aztertuko delarik, gehienbat—.

3.1 Datu multzoa sortzen

Lehen esan bezala, datu multzoko instantzia bakoitzak bost elementu ditu, bi irudi, irudi hauen goiburukoak eta goiburukoen arteko antzekotasun semantikoa deskribatzen duen balioa, $[0,5]$ tartekoa dena. Hasiera batean guztira 829 instantziako *vSTS v1.0* datu multzoa sortu zen, bi datu multzo ezberdinetako esaldi eta irudi pareak erabiliz.

- **2014-ko azpimultzoa**: *PASCAL VOC-2008* datu multzoko instantzien proportzio bat erabili da 374 instantziaz osatutako azpimultzoa eraikitzeko [43].
- **2015-ko azpimultzoa**: *Flickr-8k* datu multzoko 455 instantziaz osatutako azpimultzoa.

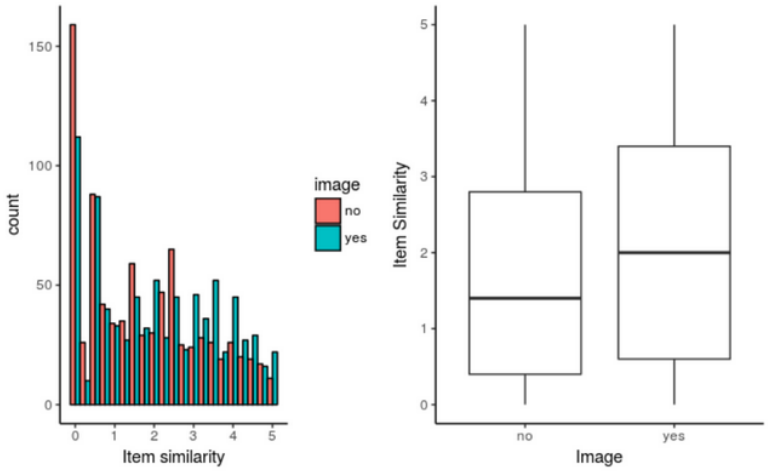
Datu multzo honekin irudiek ematen duten informazioarekin batera bi goiburukoren arteko antzekotasun semantikoa hobeto estimatzen dela ikusi nahi zen —[5] artikuluan hainbat eredu ezberdin erabiliz antzeman zena—. Gainera, hainbat pertsona datu multzoaren instantzien antzekotasuna estimatu zuten, irudien laguntzarik gabe lehenengo, eta irudiekin batera ondoren.

	Batez-bestekoa	**Mediana**	**Desbideraketa-estandarra**	**Zero kop.**
Irudirik gabe	1.72	1.40	1.46	%19.17
Irudiekin	2.10	2.00	1.52	%13.51

3.1 Taula: Irudien influentziarik gabeko eta influentziarekin lortutako estimazioen batez-bestekoa, mediana, desbideraketa-estandarra eta zeroko balio kopuruak.

3.1 taulan bi kasuetan lortutako estimazioen batez-bestekoa, mediana, desbideraketa-estandarra eta zeroko balio kopuruak azaltzen dira, 3.1 grafikoan estimazioak irudikatzen direlarik.

Azkeneko irudiko kaxa-diagrametan lortutako emaitzak orekatuagoak direla ikus daiteke; baina, bi kasuetan, *vSTS v1.0* datu multzoko azpimultzo handi bat antzekotasun baxuko instantziez osatuta dagoela antzematen da. Desoreka hau datu multzoa hedatzerakoan saihestu nahi da.



3.1 Irudia: Bi estimazio ezberdinetan lortutako antzekotasunak. Gorriz, irudien laguntzarik gabe lortutakoak, eta urdinez besteak.

3.2 Datu multzoa hedatzen

vSTS v1.0 datu multzoa hedatzen hasi baino lehen, hedapena egin ondoren lortutako datu multzoak bete behar dituen hainbat ezaugarri definitu ziren. Alde batetik, datu multzo orekatu bat eraiki nahi zen. Izan ere, aurretik burututako esaldien aukeraketan hautatutako esaldi pareek antzekotasun semantiko baxua izateko tendentzia zutela ikusi zen. Hau ez da harritzekoa, ausaz bi esaldi aukeratzen badira, beraien artean antzekotasunik edukitzeak probabilitate txikia du eta. Beste aldetik, irudietan barietatea egotea nahi zen, hots, instantzia ezberdinetan irudi parerik ez errepikatzea eta irudien dibertsitatea kontuan hartzea; non kasu batzuetan antzekotasun semantikoa zuzen estimatzeko irudi eta goiburukoek ematen duten informazio konbinatua beharrezkoa den.

Funtsean, hedapen honetako instantziak aurreko kapituluan definitu diren *Flickr30k* eta *MS-Coco* multzoetatik hartu dira. Behin instantzien hautaketa egin ondoren, esaldi pare horien antzekotasun semantikoak eskuz idatzi dira, *Amazon Mechanical Turk* edo *AMT* plataforma erabiliz.

AMT ikertzaile eta enpresen bizitza laguntzeko pentsatuta dago, birtualki burutu daitez-

keen ataza sinpleak publikatzeko *API* bat eskura jartzen duelako. Bertan, ataza horiek burutzeko prest dauden pertsonek edo *Turker*rek ataza horiek betetzen dituzte, egin duten lan kopuruaren arabera ikertzaileek jarri duten dirutik kobratzen dutelarik.

3.2.1 Instantzien laginketa

vSTS datu multzoa eraikitzeko aurrekontu finitu bat eskura zegoenez, hasiera batean *AMT* plataformara igotzeko 3000 instantziako muga bat ipini zen. Hortaz, lehenik eta behin 3000 instantzia horiek aukeratu behar ziren. Hala ere, aukeratzen den instantzia multzoa orekatua dagoen ala ez jakiteko, instantzien antzekotasun semantikoen balioak ezagutu beharko lirateke —printzipioz, *AMT* plataforman ezarri behar dira—. Arazoa, beraz, antzekotasun semantikoa aldez aurretik ezagutu edo estimatzea da, *AMT* plataformara datu multzo orekatu bat igotzeko.

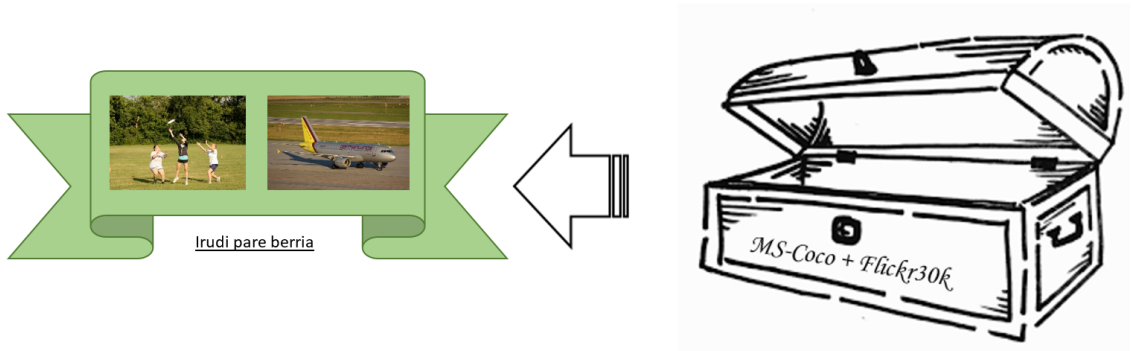
Antzekotasun semantikoak $[0,5]$ tarteko balioekin adierazten direnez, tarte hori bost azpitartetan banatu daiteke: $[0,1],(1,2],\ldots,(4,5]$. Datu multzoa orekatua mantentzeko tarte bakoitzean gehienez 600 instantzia inguru egon beharko lukete. Beraz, *AMT* plataformara igo baino lehen, instantzia horien antzekotasun semantikoak estimatu egin ziren hainbat tresna erabiliz —3.2 taula—.

Overlap	Hitz-zaku edo *BoW* eredua jarraituz esaldietan agertzen diren hitz ezberdinen agerpenak bektore bitar batean kodetzen dira + kosinuaren-antzekotasuna.
Caverage	Esaldi bakoitzaren *GloVe* hitz-bektoreen zentroideak + kosinuaren-antzekotasuna.
Image	*ResNet50* sistema aurre-entrenatuaren azken-aurreko kaparen balioak + kosinuaren-antzekotasuna.
ML	*Machine Learning* sistemetan —gehienbat itzulpen automatikoan eta distantzietan— oinarritutako ezaugarri multzoa. Goiburuko azpimultzoa ez duen *STS-B* datu multzoan entrenatu da —5. kapituluan aipatzen den *STS-B** multzoan—.

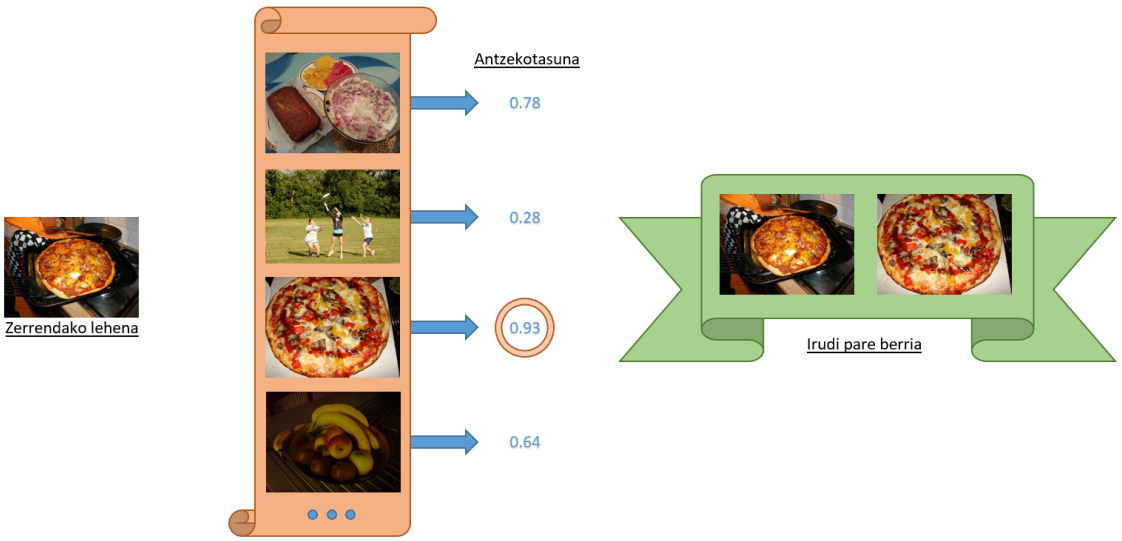
3.2 Taula: Antzekotasuna neurtzeko lau metodo hauen deskribapenekin batera, sinpleenetik konplexuenera ordenatuta daudelarik. *Image* metodoa irudiak erabiltzen dituen bakarra da.

Instantziak sortzerako orduan lehenik eta behin irudi pareak aukeratu ziren. Esan bezala, irudi pare horiek *Flickr30k* eta *MS-Coco* multzoetatik hautatu ziren —guztira 150K irudi baino gehiago kontuan hartuz—. Irudi bakoitzak bost goiburuko dituenez, goiburukoen hautaketa esaldi pareek partekatzen zituzten hitz kopuruak erabili ziren —*Overlap* metodoa—.

Irudien laginketarekin hasiz, ataza honetarako bi laginketa mota ezberdin erabili ziren: ausazko laginketa eta irudien antzekotasun bidezko laginketa —3.2 irudia—. Laginketa mota bakoitzean 1500 instantzia sortzeko muga ezarri zen —300 instantziako muga jarriz laginketa bakoitzeko azpi-tarte bakoitzean—.



(a)



(b)

3.2 Irudia: Irudi pareen laginketa motak: (a) ausazko laginketa eta (b) irudien antzekotasuna bidezko laginketa, *Image* metodoa erabiliz lortzen dira irudi bakoitzaren antzekotasunak.

Ausazko laginketan aztertu ziren 155068 irudietatik 77534 pare sortu ziren, irudi bakoitza behin bakarrik hautatu zelarik. Aukeratutako irudi pareak oso ezberdinak zirenez, goiburuko pareak ausaz aukeratu beharrean *Overlap* metodoa erabili zen antzekotasun-baxuko pareak ez sortzeko ahaleginean —3.3 irudia—.

Irudien antzekotasun bidezko laginketan irudi pare kopuru bera hautatu zen, aurreko kasuan bezala irudi bakoitza behin hautatuz. Prozesu honetan irudiak zerrendatuta izanik,

A. Three young women are trying to catch a frisbee.
B. A group of woman standing in a field trying to catch a frisbee
C. Three girls playing frisbee together in a field.
D. 3 people attempt to catch a frisbee in midair.
E. three people in a grass field reaching for a frisbee in the air

1. A modern jet aircraft on the runway of an airport
2. A german airplane is on an airport runway.
3. A lone aircraft is sitting on a long runway.
4. a silver plane sits on an airport tarmac
5. A plane that is on the ground in the day light.

	1	2	3	4	5
A	a	a	a	a	a
B	a, of	a	a	a	a, in
C	a	a	a	a	a, in
D	a	a	a	a	a, in
E	a, the	a	a	a	a, in, the

3.3 Irudia: *Overlap* metodoa erabiliz 25 esaldi pareen artetik bat aukeratzen da. Kasu honetan *Overlap* metodoarekin lortzen den esaldi pareak hiru hitz partekatzen ditu, bi esaldien *one-hot* kodeketen arteko kosinuaren antzekotasuna handiena du eta.

zerrendako lehenengo irudia bere irudi antzekoenarekin parekatzen da, *Image* metodoa erabiliz. Ondoren, prozesua errepikatu egiten da, bi irudi horiek zerrendatik kenduz zerrenda hutsik geratu arte.

Behin irudi eta esaldi pareak aukeratuta zeudenean, haien antzekotasun semantikoa estimatzeko saiakera egin zen 3.2 taulako metodoekin. Metodo hauek tarte jakin bateko balioak itzultzen dituztenez, balio horiek $[0,5]$ tartera eskalatu ziren —3.3 taula—.

Behin 3.3 taulako datuak izanik, instantziak hautatzeko irizpidea definitu behar izan zen. Kontuan izan behar da *vSTS* datu multzoaren hedapena burutzen ari zela; eta, hortaz, 829 instantzia bere antzekotasunen balioekin eskura zeudela. Hori dela eta, 3.3 taulako tresnak erabiliz *vSTS v1.0* datu multzoaren gainean estimazioak burutu ziren. Ondoren, estimazio horiek datu errealekin konparatu ziren *Pearson*en korrelazioa erabiliz.

$$\rho_{\mathbf{x},\mathbf{y}} = \frac{\mathrm{cov}(\mathbf{x},\mathbf{y})}{\sigma_{\mathbf{x}}\sigma_{\mathbf{y}}} \tag{3.1}$$

Sim	Overlap	Caverage	Image	ML
0	52393	0	0	1104
(0,1]	16328	65	10	64725
(1,2]	7758	5108	2801	9584
(2,3]	940	33521	60654	1858
(3,4]	106	33262	14069	252
(4,5]	9	3578	0	11

Sim	Overlap	Caverage	Image	ML
0	17724	0	0	175
(0,1]	16793	8	0	24073
(1,2]	22397	423	4	21316
(2,3]	14152	7720	932	19645
(3,4]	5313	37672	73423	11076
(4,5]	1155	31711	3175	1249

3.3 Taula: Laginketa ezberdinekin lortutako instantzien antzekotasunak dira. Lehenengo taula ausazko laginketari dagokio; eta bigarrena, berriz, beste laginketari.

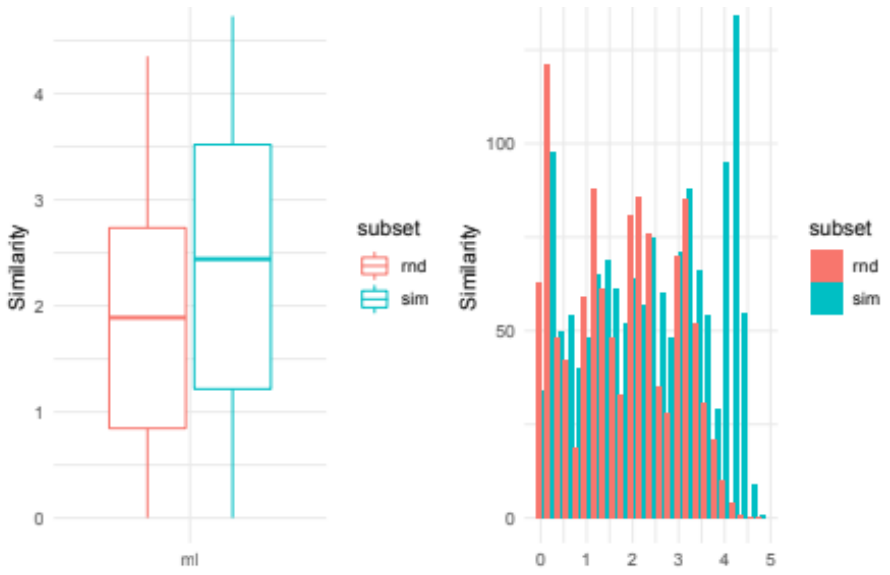
3.1 ekuazioan $\mathbf{x}$ eta $\mathbf{y}$ bektoreen ρ korrelazioa nola kalkulatzen den azaltzen da, *cov* kobariantza —bi aldagai kuantitatiboren arteko korrelaziorako neurria— eta $\sigma_{\mathbf{x}}$ desbideraketa-estandarra izanik. Geroz eta korrelazio altuagoa, orduan eta hobeagoak dira estimazioak. Korrelazio horiek 3.4 taulan azaltzen dira.

Korrelazio hauek nahiko altuak dira, *Overlap* eta *ML* ereduetan batez ere. *ML* ereduak korrelazio altuena duenez, hau erabili zen instantziak aukeratzeko orduan, hau da, antzekotasunen azpi-tarte eta laginketa bakoitzeko 300 instantzia ausaz hautatzerakoan. Ausazko laginketan *ML* eredua azpi-tarte batzuetan ez da 300 instantziara iristen. Horregatik, bukaera batean 2639 instantzia bakarrik aukeratu ziren.

Eredua	Pearson
Overlap	0.80
Caverage	0.75
Image	0.61
ML	0.85

3.4 Taula: Eredu ezberdinen errendimendua *vSTS v1.0* instantzietan, *Pearson* korrelazioen bidez neurtua.

ML ereduak kalkulatutako instantzien antzekotasun semantikoen estimazioak 3.4 irudian azaltzen dira. Oro har, estimazioen arabera datu multzo orekatu bat eraiki zela ikus daiteke.



3.4 Irudia: Laginketa bakoitzean lortutako instantzien antzekotasunak, *ML* ereduak neurtua.

3.2.2 Instantzien zailtasuna

Laginketa burutzean 2639 instantziak *AMT* plataforman ipini ziren. Instantzia bakoitzaren antzekotasuna bost pertsonek aztertu behar zuten, instantziaren antzekotasun semantikoa estimazio horien batez-bestekoa kalkulatuz.

Kontuan izan behar da *Turker* hauek sei aukera zituztela instantzien antzekotasunak esleitzerako orduan: 0, 1, 2, 3, 4 eta 5 balioak alegia[1]. Balio bakoitzak bere esanahia dauka, 2. kapituluan azaltzen den 2.2 taulan definitzen direnak. Hori bai, *Turker*ren estimazioen batez-bestekoa egitean, behin-betiko antzekotasunetan zenbaki hamartarrak ager daitezke.

	Batez-bestekoa	**Mediana**	**Desbideraketa-estandarra**	**Zero kop.**
Estimazioak	1.96	1.8	1.65	%20.8
Desadostasunak	0.6	0.55	0.45	%27.4

3.5 Taula: *Turker*rek egin dituzten antzekotasunen estimazioen batez-bestekoak, medianak eta desbideraketa-estandarrak datu multzo orekatu bat dela esaten digu. Instantzia bakoitzean batez-beste 0.6 puntuko desadostasuna egon da *Turker*ren artean.

Guztira 55 *Turker*rek parte hartu zuten, bakoitzak batez-beste 220 instantziaren antzekotasuna estimatuz. *Turker* batek besteen estimazioekiko korrelazio baxuak zituenez, $\rho < 0.75$

[1] Murrizketa hau dela eta, antzekotasun balioak zenbaki arruntetan multzokatzen dira.

izanik, pertsona honen estimazioak ez ziren kontuan hartu batez-bestekoak kalkulatzean. *Turker*ren estimazioen laburpen bat 3.5 taulan aurki daiteke.

2639 instantzietatik bostenak zeroko antzekotasuna izatea kontuan hartzekoa da; baita *Turker*ren emaitzetako %27.4-etan desadostasunik ez egotea ere —hau da, bost pertsonek balio bera esleitzea—. Oro har, datu multzoa orekatuta badago ere, antzekotasun baxuko instantziak sarriago azaltzen dira.

Eredua	**Pearson**
Overlap	0.828
Caverage	0.738
Image	0.6
ML	0.848

3.6 Taula: Eredu ezberdinen errendimendua anotazio berrietan, *Pearson* korrelazioen bidez neurtua.

3.6 taulako emaitzak *Overlap* eta *ML* metodoen errendimendu oso ona frogatzen badute ere —antzekotasunarekiko 0.828 eta 0.848ko korrelazioak dituzte eta—, sortu den datu multzoa errazegia dela esan daiteke. Izan ere, *Overlap* bezalako eredu sinple batek 0.828ko korrelazio altua lortzea hobekuntzak lortzeko marjina txikia uzten du.

Overlap metodoaren errendimendu altuak arrazoi bat izan dezake. Izan ere, esaldi pareak hautatzerakoan metodo bera erabili zen, instantzien antzekotasuna kondizionatu zezakeelarik.

Datu multzoan erraztasun hau zuzendu ahal izateko instantzia errazak detektatzeko hainbat modu definitu eta beste hainbat proba ezberdin erabili baziren ere, datu multzoko instantzia errazenak detektatzeko erabili zen prozesu definitiboa bakarrik deskribatuko da, hainbat proba egin ondoren modu zuzenena kontsideratu zena. Hortaz, 3.3 ekuazioak instantzia baten zailtasuna definitzen du, lagin-multzo baterako *Pearson*en korrelazioan oinarrituta —3.2 ekuazioan—.

$$\rho_{x,y} = \frac{1}{n-1} \sum_{i=1}^{n} \left(\frac{x_i - \bar{x}}{\sigma_x} \right) \left(\frac{y_i - \bar{y}}{\sigma_y} \right) \tag{3.2}$$

Notazioa aipatzearren, $\bar{x}$ **x** lagineko elementuen batez-bestekoa adierazten du, n instantzia kopurua izanik. 3.3 ekuazioan $\overline{\text{sim}}$ eta σ_{sim} balioak jada azaldu dira, 3.5 taulan agertzen diren antzekotasunen bataz-besteko eta desbideraketa-estandarrak baitira, hurrenez hurren. 3.3 ekuazioko irizpidea erabiliz lortzen diren balioak handiagoak diren heinean, ekuazioan erabiltzen ari den lagina errazagoa dela esan daiteke.

$$\text{Erraztasuna} = \left(\frac{\text{overlap}_i - \overline{\text{overlap}}}{\sigma_{\text{overlap}}}\right)\left(\frac{\text{sim}_i - \overline{\text{sim}}}{\sigma_{\text{sim}}}\right) \tag{3.3}$$

Ondoren, zailtasun metrika hau erabiliz, instantzia guztiak zailenetik errazenera ordenatu ziren; eta 20 azpimultzo ordenatuetan banatu ziren —lehenengo $\mathbf{b}_1$ azpimultzoak datu multzoko instantzia zailenak zituelarik, eta azkeneko $\mathbf{b}_{20}$ azpimultzoak errazenak—, azpimultzo bakoitzak instantzia guztien %5-a izanik. Ondoren, aldez aurretik definitutako lau metodoekin azpimultzo horietako estimazioak eta hauen *Pearson* korrelazioak kalkulatu ziren, korrelazio horiek 3.7 eta 3.8 tauletan azaltzen dira.

	$\mathbf{b}_1$	$\mathbf{b}_2$	$\mathbf{b}_3$	$\mathbf{b}_4$	$\mathbf{b}_5$	$\mathbf{b}_6$	$\mathbf{b}_7$	$\mathbf{b}_8$	$\mathbf{b}_9$	$\mathbf{b}_{10}$
Overlap	-0.79	-0.71	-0.60	-0.56	-0.44	-0.31	-0.16	-0.01	0.12	0.25
Caverage	-0.46	-0.33	-0.20	-0.19	-0.11	-0.05	0.03	0.12	0.21	0.30
Image	0.71	0.62	0.58	0.52	0.49	0.48	0.48	0.46	0.44	0.47
ML	-0.58	-0.44	-0.29	-0.24	-0.12	-0.03	0.10	0.23	0.33	0.42

3.7 Taula: Lau metodoekin kalkulatu diren hamar azpimultzo zailenen korrelazio agregatuak, non $\mathbf{b}_3$ zutabeko korrelazioak $\mathbf{b}_1 \cup \mathbf{b}_2 \cup \mathbf{b}_3$ azpimultzoari dagozkion.

	$\mathbf{b}_{11}$	$\mathbf{b}_{12}$	$\mathbf{b}_{13}$	$\mathbf{b}_{14}$	$\mathbf{b}_{15}$	$\mathbf{b}_{16}$	$\mathbf{b}_{17}$	$\mathbf{b}_{18}$	$\mathbf{b}_{19}$	$\mathbf{b}_{20}$
Overlap	0.35	0.43	0.51	0.57	0.64	0.68	0.71	0.75	0.79	0.83
Caverage	0.38	0.44	0.48	0.52	0.57	0.61	0.64	0.67	0.70	0.74
Image	0.49	0.49	0.50	0.49	0.49	0.50	0.51	0.53	0.57	0.60
ML	0.50	0.57	0.62	0.66	0.70	0.74	0.76	0.79	0.82	0.85

3.8 Taula: Lau metodoekin kalkulatu diren hamar azpimultzo errazenen korrelazio agregatuak, non $\mathbf{b}_{12}$ zutabeko korrelazioak $\mathbf{b}_1 \cup \mathbf{b}_2 \cup \ldots \cup \mathbf{b}_{12}$ azpimultzoari dagozkion.

Azkeneko taula hauetan agertzen diren korrelazioetan testuetan oinarritzen diren metodoak azpimultzo zailenekin zailtasun handiak dituztela ikus daiteke. *Image* metodoak, ordea, ez du hainbesteko arazorik korrelazio txukun bat emateko azpimultzo horietan. Datu multzoaren %70 zailena hartzen bada, metodoek ematen dituzten korrelazioak nahiko baxuak direla ikus daiteke —3.9 taula—.

	%70	**%100**
Overlap	0.57	0.83
Caverage	0.52	0.74
Image	0.49	0.60
ML	0.66	0.85

3.9 Taula: Datu multzoko %70 instantzia zaileneko eta multzo osoko korrelazioak.

Datu multzoaren %70 zailenarekin geldituz, ebatzi nahi den *vSTS* atazarekin emaitza onak lortzea zaildu da. *Overlap* eta *ML* metodoen korrelazioak 0.26 eta 0.19 puntutan jaitsi dira, hurrenez hurren; ataza ebazteko metodo konplexuagoak erabiltzea bultzatzen delarik —modalitate anitzeko adierazpenak erabiltzen dituzten metodoak, adibidez—.

Beraz, hasieran definituta zegoen eta 829 instantzia bakarrik zituen *vSTS v1.0* datu multzoari gehitu zaizkion instantziak kontuan harturik, *vSTS v2.0* datu multzoa eraiki zen. Azkeneko bertsio honek 2677 instantzia ditu eta bertsio honen ezaugarriak hurrengo azpiatalean definitu dira.

3.3 Ezaugarriak

Aurretik esan bezala, datu multzo hau hainbat etapetan zehar eraiki da, lehenengoa [5] artikuluan aipatzen delarik. *STS-B* datu multzoaren antzekoa da, baina esaldiak hainbat jatorri ezberdinetatik hartu beharrean —berriak eta foroak, besteak beste—, instantzia guztiak goiburukoez eta hauen irudiez osatuta dago, *vSTS* datu multzoan aurki daitezkeen esaldietan barietate txikiagoa aurkituz.

2677 instantzia ditu, guztira 5554 esaldi eta irudi izanik. *STS-B* multzoan bezala, aurki daitezkeen instantziak hiru azpimultzo disjuntutan banatu dira: *train*, *dev* eta *test*, datu multzoak 3.10 taulan agertzen den distribuzioa duelarik.

	Train	**Dev**	**Test**	**Guztira**	**Train**	**Dev**	**Test**	**Guztira**
MS-Coco	717	347	345	1409	%26.76	%12.97	%12.88	%52.61
Flickr30k	221	114	104	439	%8.25	%4.26	%3.89	%16.39
vSTS v1.0	400	208	221	829	%14.98	%7.76	%8.25	%30.98
Guztira	1338	669	670	2677	%49.99	%24.99	%25.02	%100.0

3.10 Taula: Taula hauetan azpimultzo bakoitzak duen instantzia kopuru eta proportzioak aurki daitezke, hurrenez hurren. Instantzia bakoitzak bi esaldi ditu; beraz, esaldi kopurua tauletako balioen bikoitza da.

Lehenago *vSTS* multzoan aurki daitezkeen esaldietan barietate txikiagoa dagoela esan da, goiburukoez osatuta dago eta. Honako hau hitz-zakuak erabiliz hobeto ikusi daiteke. Ikus daitekeenez, 3.11 taulan agertzen diren hitz-zakuak *STS-B* multzoko *Caption* azpimultzoko hitz-zakuen parekoak dira, *News* eta *Forum* azpimultzoenetik asko ezberdinduz.

Datu multzo honetan aurki daitezken hitzak oso generikoak dira, izen berezien agerpena oso arraroa da eta. Gainera, *STS-B* multzoko *Caption* azpimultzoko ondorengo egitura

Train		Dev		Test	
All	without SW	All	without SW	All	without SW
a: 1.9155	man: 0.2029	a: 1.9305	man: 0.1988	a: 1.8694	man: 0.1657
in: 0.438	white: 0.1278	in: 0.4185	white: 0.1495	in: 0.4597	white: 0.1269
on: 0.3711	sitting: 0.1114	the: 0.3871	woman: 0.1031	on: 0.3761	sitting: 0.1142
the: 0.3595	woman: 0.1099	on: 0.3677	sitting: 0.1009	the: 0.3716	standing: 0.1037
of: 0.3277	black: 0.1024	of: 0.3341	dog: 0.0957	of: 0.344	black: 0.1022
and: 0.2653	two: 0.0983	and: 0.2952	black: 0.0942	and: 0.247	two: 0.1022
with: 0.2347	standing: 0.0923	with: 0.2235	two: 0.0942	with: 0.2254	people: 0.0933
man: 0.2029	people: 0.0874	is: 0.2025	standing: 0.0874	is: 0.1761	dog: 0.0881
is: 0.1928	dog: 0.0845	man: 0.1988	people: 0.074	man: 0.1657	woman: 0.0821
white: 0.1278	next: 0.065	white: 0.1495	shirt: 0.0695	white: 0.1269	next: 0.0769
sitting: 0.1114	field: 0.0635	woman: 0.1031	next: 0.0605	to: 0.1157	table: 0.0694
woman: 0.1099	street: 0.0605	sitting: 0.1009	field: 0.0598	sitting: 0.1142	group: 0.672
to: 0.1031	group: 0.0579	dog: 0.0957	wearing: 0.0598	standing: 0.1037	water: 0.059
black: 0.1024	shirt: 0.0531	black: 0.0942	holding: 0.059	black: 0.1022	street: 0.0582
two: 0.0983	table: 0.0523	two: 0.0942	group: 0.0583	two: 0.1022	field: 0.053
standing: 0.0923	holding: 0.0482	to: 0.0912	street: 0.0538	people: 0.0933	blue: 0.0485
people: 0.0874	wearing: 0.0475	standing: 0.0874	table: 0.0501	dog: 0.0881	playing: 0.0478
dog: 0.0845	blue: 0.0475	at: 0.077	red: 0.0493	woman: 0.0821	red: 0.0478
are: 0.074	red: 0.0463	people: 0.074	playing: 0.0456	are: 0.0806	wearing: 0.0478
at: 0.0714	brown: 0.0433	shirt: 0.0695	young: 0.0448	at: 0.0776	front: 0.0463

3.11 Taula: Zutabe bakoitzaren goiburukoetan *Train*, *Dev* eta *Test* azpiatal bakoitzeko hitz guztiak kontuan hartu ala hitz hutsak —*stop words* edo *SW*— kendu diren ikus daiteke. Zutabe bakoitzean azpiatal horietan gehien errepikatzen diren 20 hitzak azaltzen dira, non dagokien balioa esaldi bakoitzeko batez-besteko agerpen kopurua den.

bera erabiltzen da *vSTS*-ko ia esaldi guztietan —koloreztatuta dauden hitzak ordezkatu behar direnak dira, eta parentesi artekoak kasu gehienetan agertzen dira—.

A subject (is) verb+(ing) something.
Subjects (are) verb+(ing) something.

Hau erraz ikus daiteke 3.11 taulako hitz-zakuetan. Adibide bat jartzeagatik, *vSTS v2.0*-ko azpimultzoetan *a* hitza batez-beste $[1.86, 1.93]$ aldiz agertzen dela ikus daiteke esaldi bakoitzean, hitz honen agerpen kopurua *STS-B* multzokoa baino nahiko altuagoa izanik. Beste aldetik, *vSTS* datu multzoan subjektu pluralak aurkitzea *STS-B* multzoan baino arraroagoa dela ikus daiteke —*people* eta *are* bezalako hitzen agerpena murritzagoa da eta—.

Train, *Dev* eta *Test* azpimultzoen arteko hitz-zakuetan ez dago ezberdintasun handirik, agerpen gehienak dituzten hitzak berdinak baitira ia kasu guztietan. Egitura eta hitz ber-

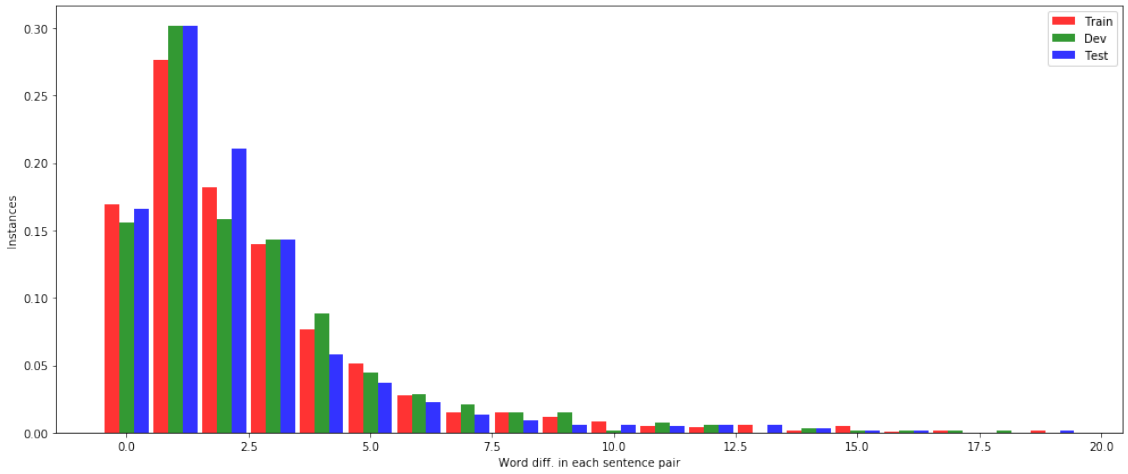
dinak erabiltzeak erabilitako neurona-sareei ataza errazten die, eta hau esperimentuak egitean kontuan hartu behar da.

Hitzen agerpen kopurua alde batera utziz, esaldien luzerak hauen konplexutasuna adierazi dezakete ere bai. 3.12 taulan ikusi daitekeenez, hiru azpimultzoen esaldi luzerek distribuzio bera jarraitzen dute. Hala ere, *STS-B*-ko *Caption* azpimultzoko esaldiak bizpahiru hitz motzagoak dira batez-beste —2.5 taulan begiratu konparazioa egiteko—; esaldiek, oro har, egitura konplexuagoak edota deskripzio gehiago dituztela esan nahi duelarik.

	Mean	**Std**	**Min**	**25%**	**50%**	**75%**	**Max**
Train	10.846	3.592	4	9	10	12	38
Dev	10.871	3.861	4	9	10	12	51
Test	10.749	3.658	4	9	10	12	45

3.12 Taula: *vSTS* datu multzoko esaldien hitz kopuruak, hiru azpimultzoetan banatuta. Ezkerretik eskuinera, hitz kopuruen batez-bestekoa, desbideraketa-estandarra, minimoa, lehen kuartila, mediana, bigarren kuartila eta maximoa.

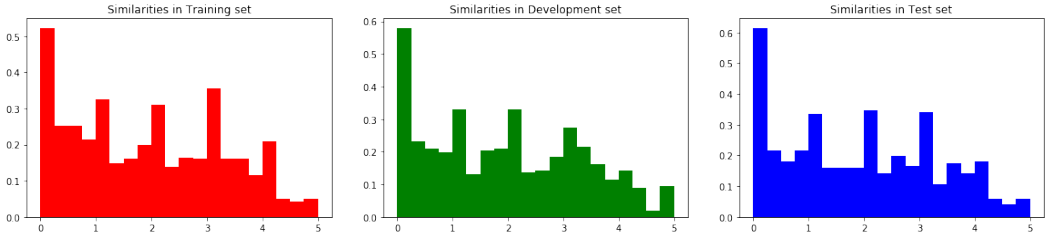
Beste aldetik, esaldi pareen luzerak ezberdinak izan daitezke. Hitz kopuru horien diferentziak hainbat sistemen errendimendua jaistea ekar dezake, *Transformer*ren irteera gerutzako token kopurua alda dezakete eta, adibidez. Kontuan izan behar da erabiltzen ari den tokenizazioaren arabera token kopuru hori ere aldatzen dela, baina hitz kopuruen diferentziak errealitatean gertatuko diren ezberdintasunen parekoak izango direla suposatzen da azterketa hau egiteko orduan —3.5 irudia—.



3.5 Irudia: *vSTS* datu multzoko esaldi pareen hitz kopuruen diferentziak, hiru azpimultzoetako instantziak ezberdinduz. Ardatz bertikalean azpiatal bakoitzaren proportzioak definitzen dira, eta horizontalean, berriz, instantzia bakoitzaren esaldien hitz kopuru diferentzia.

STS-B-ren kasuan bezala, esaldi-pareen hitz kopuru ezberdinak nahiko orekatuta daude, salbuespen batzuk kontuan hartu gabe.

vSTS v2.0-ren analisiarekin bukatzeko datu multzoak dituen instantzien antzekotasun balioak aztertuko dira. Aurrez aldetik aipatutako sistemak entrenatzerako orduan, datu multzoan estimatu nahi diren balioek distribuzio uniforme edo orekatu bat jarraitzea oso garrantzitsua da, hots, 0 eta 5 arteko antzekotasun semantikoak dituzten instantziak uniformeki zabaltzea tarte horretan. 3.6 irudi multzoan eta 3.13 taulan, aldez aurretik definitutako 3 azpimultzoetan agertzen diren instantzien antzekotasun balioak azaltzen dira.



3.6 Irudia: *vSTS v2.0* datu multzoko instantzien antzekotasun semantikoen balioak. Instantzien balioak 3 azpimultzotan banatu dira: *Train-set*, *Dev-set* eta *Test-set*.

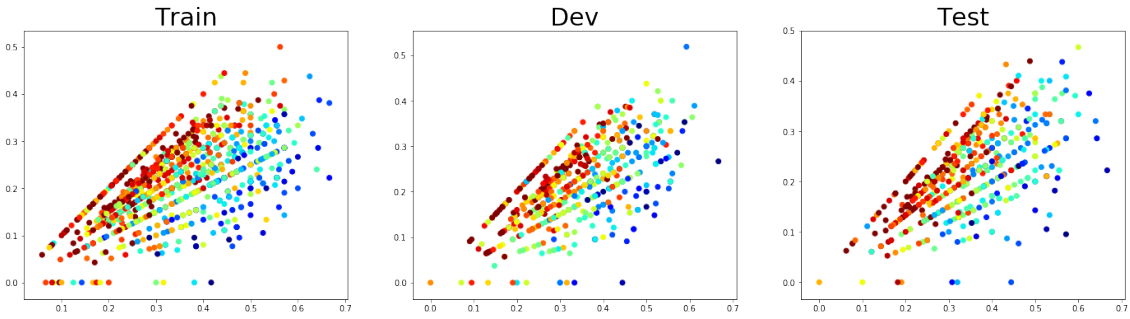
	Mean	**Min**	**25%**	**50%**	**75%**	**Max**	**Zero Kop.**
Train	1.894	0	0.6	1.8	3	5	%12.03
Dev	1.899	0	0.6	1.8	3	5	%11.95
Test	1.884	0	0.6	1.8	3	5	%14.02

3.13 Taula: *vSTS* datu multzoko instantzien antzekotasun-balioen laburpena, hiru azpimultzoetan banatuta. Ezkerretik eskuinera, hitz kopuruen batez-bestekoa, minimoa, lehen kuartila, mediana, bigarren kuartila, maximoa eta zero kopuruak agertzen dira.

Oro har, antzekotasun semantikoen balioak nahiko orekatuta daude, hau da, balio guztietarako instantzia nahikoak daude. $[0,5]$ tarteko zenbaki naturaletan tontor batzuk aurkitzen dira 3.6 irudi-multzoko grafiko guztietan. Izan ere, antzekotasun semantikoak balio errealekin adierazten badira ere, balio horiek zenbaki arrunten bidezko batez-bestekoekin kalkulatu dira —*STS-B* datu multzoan bezalaxe—.

Azkenik, 3.7 irudian antzekotasun semantikoak bi esaldiek konpartitzen dituzten hitzen proportzioekin duen zerikusia aztertzen da.

Berriz ere, *STS-B*-ko hainbat azpimultzoetan bezala. bi esaldiek partekatzen dituzten hitzen kopurua handitzen den heinean, antzekotasun semantikoa handitzen joan ohi da. Baina, hitz horiek gehienbat hitz hutsak direnean, antzekotasuna asko txikitzen da —konpartitzen dituzten hitzek ez dituztelako esanahi semantikorik—.



3.7 Irudia: vSTS v2.0 multzoko hiru azpimultzoen instantzien antzekotasunak. Grafiko bakoitzean ardatz horizontalak instantzia bakoitzeko bi esaldiek konpartitzen dituzten hitzen proportzioa adierazten du. Ardatz bertikalak berdin jokatzen du, hitz hutsak kontuan hartzen direlarik bakarrik. Puntu bakoitzaren koloreak instantziaren antzekotasun semantikoa adierazten du: 0, 1, 2, 3, 4 eta 5.

4. KAPITULUA

DiscoGAN for Multimodal Mapping arkitektura

Lan honen helburuetako bat *vSTS* ataza ebazten duen sistema berri bat sortzea da. Sistema berri honek irudiak eta goiburukoak erabiltzen ditu, goiburukoen arteko antzekotasun semantikoak estimatzeko helburuarekin. Izan ere, irudien informazio gehigarriek goiburuko pareen antzekotasunen analisiei eman dizkiokeen onurak aztertzeko sortu zen *vSTS* datu multzoa. Azken puntu hau sistema berri honekin betetzen den aztertu nahi izan da.

Kapitulu honetan *GAN* arkitekturan oinarritutako *DiscoGAN-M*^{3} sistema zertan datzan, arkitektura hau osatzen duten neurona-sare ezberdinen ezaugarri eta egiturak, eta hauen ikasketa-prozesua deskribatuko dira.

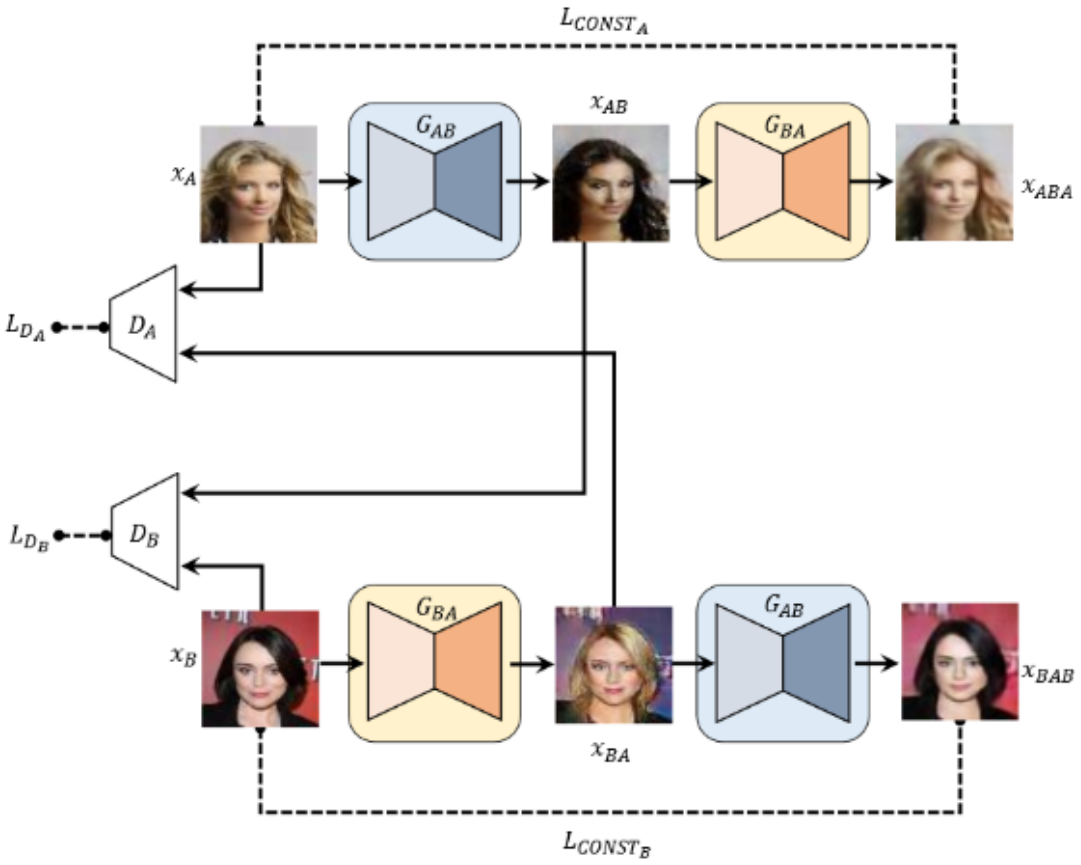
4.1 Ideia

Dokumentu honen 2. kapituluan [41] artikuluko *DiscoGAN* sistema deskribatu da, non irudietan azaltzen diren bi domeinu ezberdinen arteko erlazioak ikasteko gai den. Domeinu horien ezberdintasuna irudietan azaltzen diren pertsonen ile kolorea edo generoa, objektu ezberdinen agerpena... izan daiteke.

Sistema hau, beraz, domeinu horien mapaketa ikasteko gai da; *A* domeinu bateko irudi bat *B* domeinura transformatzeko gai dena, eta alderantziz. Hala ere, sistema mota hau ez da zertan domeinu ezberdineko erlazioak ikasteko bakarrik erabili behar. Sistemaren hainbat ezaugarri aldatuz, arkitektura honen helburua modalitate ezberdinen arteko mapaketara

aldatu daiteke —domeinu aldaketak aztertu beharrean modalitateak aldatuz, hain zuzen ere—.

Horrela, arkitektura berri honen ideia nagusia *DiscoGAN* sisteman erabiltzen diren sortzaileak irudi eta testuen arteko mapaketak burutzeko moldatzea da, hots, G_{I2T} sortzaile batek irudiak jasotzea eta hauen goiburukoak itzultzea, eta G_{T2I} sortzaileak alderantzizkoa burutzea. Oinarrizko *DiscoGAN* sisteman ez bezala, irudi eta goiburukoak parekatuta datozenez, sistema berriari probetxuzko aldaketaren bat aplikatuko zaio, honen ikasketa-prozesuan batez ere.

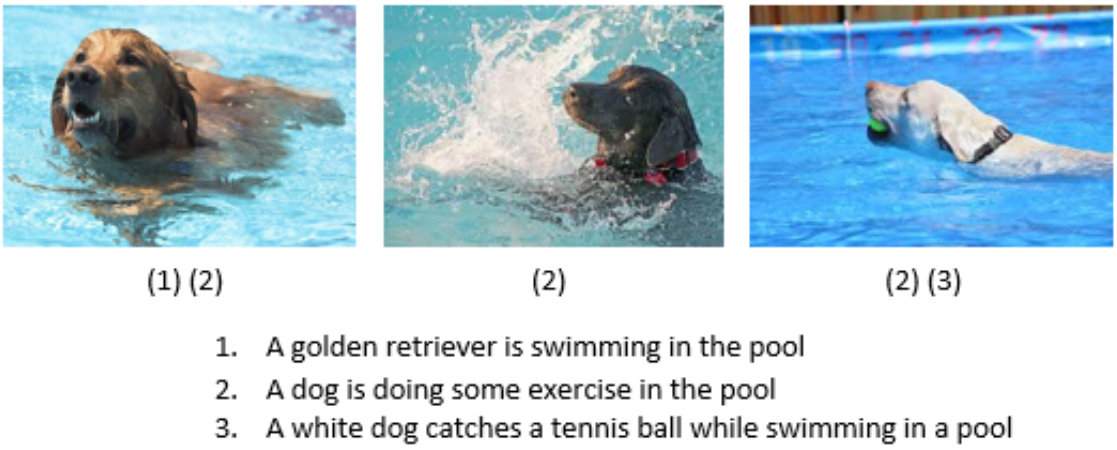


4.1 Irudia: DiscoGAN sistemaren eskema, G_{AB} eta G_{BA} sortzaileak, eta D_A eta D_B diskriminatzaileez osatuta. Kasu honetan A domeinua pertsona ilehoriez osatuta dago, eta B, berriz, ile beltzarana dutenez osatuta [41].

2. kapituluan mapaketa hauek bijektiboak izan behar direla aipatu da; hots, G_{AB}-ren kasuan B multzoko edozein $\mathbf{x}_B$ elementuri $\mathbf{x}_B = G_{AB}(\mathbf{x}_A)$ funtzioa beteko duen A multzoko $\mathbf{x}_A$ elementu bakar bat izatea gehienez —injektiboa izanik— eta A domeinuko elementu bakoitzari B multzoko elementu bat gutxienez egokitzea —suprajektiboa izanik—. Horrela, *bana-banako korrespondentzia* delakoa lortzen da, non domeinu bateko edozein

instantziak beste domeinuko pare esklusibo bat duen, lehenengo domeinuko beste instantziekin partekatzen ez duena.

Hala ere, ezaugarri hau ezin da mantendu irudi eta testuen arteko mapaketa burutzean —4.1 irudian aurki daitekeen arkitekturarekin behintzat—. Izan ere, hainbat irudi ezberdin goiburuko berdin bat erabiliz deskribatu daiteke, eta irudi bakoitza hainbat goiburukoen bidez deskribatu daiteke —4.2 irudian ikus daitekeen bezala—.



4.2 Irudia: Irudi eta esaldi pare ezberdinak eraiki dira; (2) esaldiak irudi guztiak deskribatzen dituelarik, eta lehenengo eta azkeneko irudiek bi goiburuko ezberdin izan ditzaketelarik.

Goiburuko jakin batek beste batek baino detaile gehiagorekin deskribatu ditzake irudi baten ezaugarriak, baina horrek ez du esan nahi biak egokiak ez direnik. Azkeneko gertaera hau ezin da aldatu, gauza jakin bat deskribatzeko hainbat esaldi ezberdin erabili ahal izatea hizkuntza naturalek duten ezaugarri aldaezinetako bat da eta.

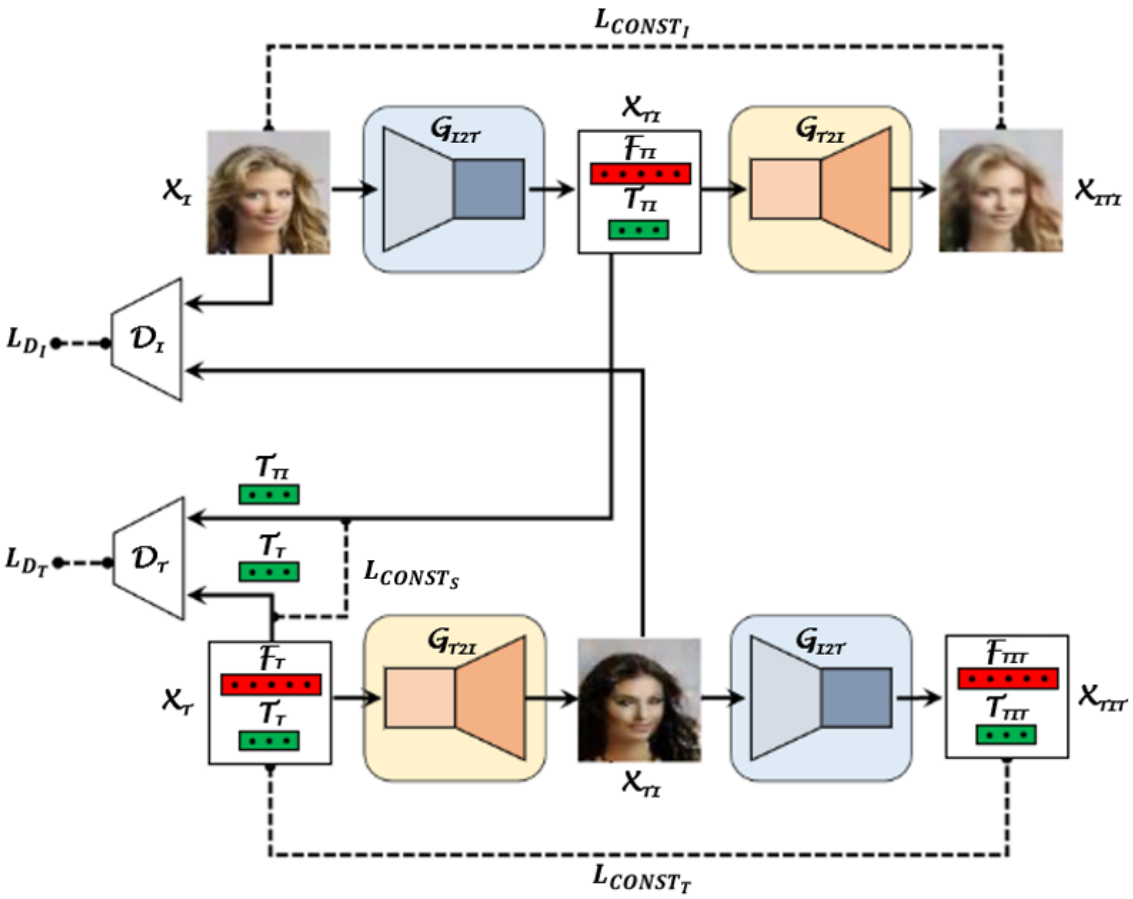
Hala ere, hainbat irudi ezberdin goiburuko berdin bat erabiliz deskribatu ahal izatearen arazoa konpontzeko, hasiera batean komentatu diren G_{I2T} eta G_{T2I} sortzaileek burutzen dituzten funtzioak eraldatu egingo dira —hurrengo azpiatalean deskribatuko dira—.

Aipatutako mapaketa hau ikasten bada, irudi eta goiburukoak espazio berean maneiatzeko gaitasuna lortuko da, modalitate anitzeko errepresentazioak ez badira ere. Horrela, kosinuaren antzekotasuna edota erregresio kapa bat gehituz, esaldi eta irudien arteko antzekotasunak neurtzeko ahalmena lortuko da, *vSTS* atazan egin daitezkeen estimazioak sendotuz.

Beste ezer esaten ez den bitartean irudiek $w \times h \times 3$ dimentsioa dutela —w irudiaren zabalera eta h altuera izanik—, eta goiburukoek adierazten dituzten bektoreak ezagunak direla eta aurre-konputatua daudela suposatuko da.

4.2 Arkitekturaren ezaugarriak

DiscoGAN sistemaren bertsio berri hau 4.3 irudian azaltzen da, eta hemendik aurrera *DiscoGAN-M*^{3} bezala adieraziko da. Adierazpen hau *DiscoGAN for MultiModal Mapping* ingelesetik dator, eta honen itzulpena "Modalitate anitzeko mapaketarako *DiscoGAN*" izango litzateke.



4.3 Irudia: DiscoGAN-M^3 sistemaren eskema. Sistema hau G_{I2T} eta G_{T2I} sortzaileek, eta D_I eta D_T diskriminatzaileek osatzen dute.

Oinarrizko *DiscoGAN* sisteman sortzaile baten aplikazioak sarrerako irudien aurkitzen den informazioa zerbait aldatzen du, irudien domeinua aldatzeko asmoarekin. Baina, kasu honetan modalitatez aldatu nahi denez, sarrera eta irteerako datuek informazioa mantendu behar dute —hauek kodetuta dauden modua bakarrik aldatuz—.

Horregatik, sarrerako irudi bat testuen modalitatera transformatzean, irudi horren informazioa bi ataletan deskonposatuko da. Alde batetik, goiburukoa kodetzen duen bektorea; eta, bestetik, sarrerako irudiaren ezaugarriak gordetzen dituen bektorea.

Hori dela eta, G_{I2T} sortzaileak ez du goiburukoaren errepresentazioa bakarrik itzuliko —hemendik aurrera goiburuko-bektore bezala adieraziko dena, irudiaren ezaugarri bisualak gordeko dituen bektore batekin lagunduta etorriko da eta.

Formalizatu dezagun, beraz, sortzaile bakoitzak burutzen duena. G_{I2T} sortzaileak $\mathbf{x}_I$ irudi bat jasotzen duenean, $\mathbf{x}_{IT}$ bektorea itzuliko du —4.1 ekuazioan definitzen dena, $\oplus$ ikurrak konkatenazioa adierazten duelarik—. $\mathbf{f}_{IT}$ jatorrizko irudiaren ezaugarri bisual eta espazialak gordetzen dituen bektorea da eta $\mathbf{t}_{IT}$, berriz, $\mathbf{x}_I$ irudia deskribatzen duen goiburukoaren bektorea.

$$\mathbf{x}_{IT} = \mathbf{f}_{IT} \oplus \mathbf{t}_{IT} = G_{I2T}(\mathbf{x}_I) \tag{4.1a}$$

$$\mathbf{x}_{ITI} = G_{T2I}(\mathbf{x}_{IT}) \tag{4.1b}$$

Aldiz, G_{T2I} sortzaileak $\mathbf{x}_{IT}$ bektorea hartzen du, jatorrizko irudia berreskuratzeko ahaleginean —4.2 ekuazioa—; hau da, G_{I2T} burutzen duen funtzioaren alderantzizkoa burutzen du.

$$\mathbf{x}_{TI} = G_{T2I}(\mathbf{x}_T) = G_{T2I}(\mathbf{f}_T \oplus \mathbf{t}_T) \tag{4.2a}$$

$$\mathbf{x}_{TIT} = \mathbf{f}_{TIT} \oplus \mathbf{t}_{TIT} = G_{T2I}(\mathbf{x}_{TI}) \tag{4.2b}$$

$\mathbf{x}_T$ familiako bektoreak —hau da, $\mathbf{x}_T$, $\mathbf{x}_{IT}$ eta $\mathbf{x}_{TIT}$ bektoreak— $\mathbf{f}_T$ irudia eta $\mathbf{t}_T$ goiburukoa deskribatzen dituzten bektoreez osatuta egoteak G_{I2T} eta G_{T2I} sortzaileek burutzen dituzten funtzioei bijektiboak izateko ahalmena ematen die, hots, bana-banako korrespondentzia delakoa ahalbidetzen du.

Izan ere, irudi ezberdin bakoitzak goiburuko bera partekatu badezake ere, irudi bakoitzak bere ezaugarriak ditu —beste irudi ezberdinetan errepikatzen ez direnak—. Hortaz, $\mathbf{x}_I$ irudi bakoitzak dagokion $\mathbf{f}_{IT}$ bektore bakarra izango du; eta $G_{I2T} : \mathbb{R}^{w \times h \times 3} \longrightarrow \mathbb{R}^{f+t}$ funtzio bijektibo bat osatuko du —f balioa $\mathbf{f}_{IT}$ bektorearen dimentsioa eta t $\mathbf{t}$ bektorearena izanik—. Kasu berean $G_{T2I} : \mathbb{R}^{f+t} \longrightarrow \mathbb{R}^{w \times h \times 3}$ funtzioa bijektiboa izango da ere bai.

Sortzaileekin konparatuz, diskriminatzaileek ez dituzte aldaketa handirik jasan. Oinarrizko *DiscoGAN* arkitekturan bi diskriminatzaile erabiltzen dira sortutako irudiak errealekin ezberdintzeko. Orain, aldiz, D_T diskriminatzailea goiburuko-bektoreak ezberdintzeko erabiliko da, D_I sarea irudiak ezberdintzeko erabiltzen delarik.

Aurreko 4.3 irudian ikus daitekenez, irudien ezaugarriak kodetzen dituzten **f** bektoreak ez dira diskriminitzaile batekin aztertzen. Izan ere, kodeketa hori jarraitzen dituzten bektoreak sortzaileek eraiki ditzakete bakarrik —ala ausaz sortu daitezke, baina ez du zentzurik hauek ausaz sortzea, irudiaren ezaugarriak gorde nahi dira eta—.

Galera-funtzioei begira aldaketa gutxi egin dira. *DiscoGAN* arkitekturan bezala, berreraikitze galera-funtzio batzuk daude, L_{CONST_I} eta L_{CONST_T}, adibidez, 4.3 ekuazioetan azaltzen direnak —d funtzioak bi aldagaien arteko distantzia kalkulatzen du, eta kasu honetan batez-besteko errore koadratikoa erabili da—.

$$L_{\text{CONST}_I} = d(G_{T2I} \circ G_{I2T}(\mathbf{x}_I), \mathbf{x}_I) = d(\mathbf{x}_{ITI}, \mathbf{x}_I) \tag{4.3a}$$

$$L_{\text{CONST}_T} = d(G_{I2T} \circ G_{T2I}(\mathbf{x}_T), \mathbf{x}_T) = d(\mathbf{x}_{TIT}, \mathbf{x}_T) \tag{4.3b}$$

Hala ere, oinarrizko *DiscoGAN*ean ez bezala, irudi eta goiburukoak parekatuta daudenez —hau da, $\mathbf{x}_T$ goiburukoa $\mathbf{x}_I$ irudiari dagokionez—, irudietatik goiburukoak lortzeko laguntzen duen galera funtzio gehigarri bat gehitu da —4.4 ekuazioan definitzen dena—.

$$L_{\text{CONST}_S} = d(\mathbf{t}_{IT}, \mathbf{t}_T) \tag{4.4}$$

Azkeneko galera-funtzioa ez da irudietara aplikatzen, hots, $d(\mathbf{x}_{TI}, \mathbf{x}_I)$ ez da minimizatzen, ikasketa-prozesuan $\mathbf{f}_T$ sarrera bektorea $\mathbf{f}_{IT}$ bektorearen ezberdina baita. Bestela, G_{T2I} bera bi aldiz burutuko litzateke aurreranzko propagazio batean, eta hori ekidin nahi da.

Berreraikitze-funtzioak alde batera utzita, diskriminatzaileen estimazioen arabera sortzaile eta diskriminatzaileek ondorengo galera-funtzioak jarraituz burutzen dute ikasketa —4.5 eta 4.6 ekuazio multzoak—.

$$L_{\text{G}_{T2I}} = -\mathbb{E}_{\mathbf{x}_T \sim P_T}[\log D_I(G_{T2I}(\mathbf{x}_T))] \tag{4.5a}$$

$$L_{\text{G}_{I2T}} = -\mathbb{E}_{\mathbf{x}_I \sim P_I}[\log D_T(G_{I2T}(\mathbf{x}_I))] \tag{4.5b}$$

$$L_{\text{D}_I} = -\mathbb{E}_{\mathbf{x}_I \sim P_I}[\log D_I(\mathbf{x}_I)] - \mathbb{E}_{\mathbf{x}_T \sim P_T}[\log(1 - D_I(G_{T2I}(\mathbf{x}_T)))] \tag{4.6a}$$

$$L_{\text{D}_T} = -\mathbb{E}_{\mathbf{x}_T \sim P_T}[\log D_T(\mathbf{x}_T)] - \mathbb{E}_{\mathbf{x}_I \sim P_I}[\log(1 - D_T(G_{I2T}(\mathbf{x}_I)))] \tag{4.6b}$$

4.2.1 Sortzaile eta diskriminatzaileen detaileak

Orain arte, sortzaile eta diskriminatzaileak kaxa-beltzak bezala tratatu dira; baina bere barrutiak ez dira oso konplexuak. Sortzaile batekin hasiz, G_{I2T} sareak *AlexNet* neurona-sarearen egitura jarraitzen du. Konboluzio-sare konplexuagoak eskura badaude ere, kontuan izan behar da konplexutasun handiago batek ikasketa-prozesua asko luzatu beharko lukeela.

Beste aldetik, *AlexNet* bezalako konboluzio-sare sinple bat erabiltzeak —*ResNet* eta *VGG* sareekin konparatuz behintzat—, sareak ataza ikasteko duen gaitasun nahikoa ez izatea ekar dezake. Hala ere, erabilitako *hardware*ak duen konputazio gaitasuna eta, gehienez, ikasketa-prozesua gehienez aste batean burutu nahi dela ikusita, aukera hau egin da.

G_{I2T} sareak bost konboluzio-geruza eta hiru geruza-dentso ditu, *pooling* geruza batzuk tartean sartuz. Geruza bakoitzaren ezaugarriak 4.1 taulan azaltzen dira.

Geruza	**Input Size**	**Output Size**	**F. Size**	**F. Count**	**Stride**	**Padding**	φ
1. Conv	227x227x3	55x55x96	11	96	4	0	ReLU
1. Pool	55x55x96	27x27x96	3		2	0	
2. Conv	27x27x96	27x27x256	5	256	1	2	ReLU
2. Pool	27x27x256	13x13x256	3		2	0	
3. Conv	13x13x256	13x13x384	3	384	1	1	ReLU
4. Conv	13x13x384	13x13x384	3	384	1	1	ReLU
5 Conv	13x13x384	13x13x256	3	256	1	1	ReLU
5. Pool	13x13x256	6x6x256	3		2	0	
6. FC	6x6x256	4096					ReLU
7. FC	4096	4096					ReLU
8. FC	4096	1536					Tanh

4.1 Taula: G_{I2T} sortzailea osatzen duten geruzak. Zutabe bakoitzak geruza bakoitzaren ondorengo ezaugarriak adierazten ditu, hurrenez hurren: geruza mota, sarrera-datuen dimentsioak, irteera-datuen dimentsioak, filtroaren tamaina, filtro kopurua, *stride*, *padding* bidez gehitutako lerro/zutabe kopurua, eta φ aktibazio-funtzioa.

2.23 irudian deskribatu den *AlexNet* sarean bezala, operazio berberak burutzen dira G_{I2T} sarean. Hori bai, azkeneko geruza aldatu egin da ataza honetan behar diren emaitzak ahalbidetzeko.

4.1 taulan ikusten denez, G_{I2T} sarea funtzio bezala definitu daiteke: $G_{I2T} : \mathbb{R}^{227\times227\times3} \longrightarrow \mathbb{R}^{1536}$, non sarrerako irudia $227 \times 227 \times 3$ tamaina eta irteerako bektoreak 1536 elemen-

tu dituen. Irteerak propietate jakin batzuk izan behar dituenez, azken geruzan aktibazio-funtzioa aldatu egin da, irteerako balioak $(-1,1)$ tartera mugatzeko.

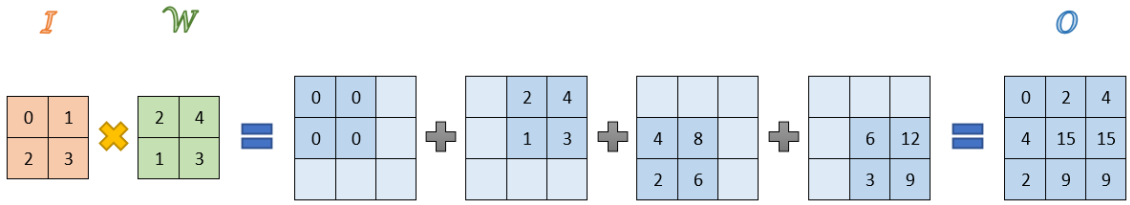
Hortaz, $\mathbf{x}_{IT} = G_{I2T}(\mathbf{x}_I)$ bektoreak 1536 balio ezberdin ditu. Balio horietatik lehenengo 1024 balioek sarrerako irudiaren $\mathbf{f}_{IT}$ ezaugarri-bektorea osatuko dute, eta beste 512 balioek $\mathbf{t}_{IT}$ goiburuko-bektorea. Izan ere, 2. kapituluan definitu den *Universal Sentence Encoder* esaldi-kodetzailea erabili da esaldien adierazpenak lortzeko, errepresentazio hauek 512 elementu dituztelarik.

Beste sortzailearekin jarraituz, G_{T2I} sortzailearen geruza bakoitza 4.2 taulan aurki daiteke. G_{T2I} sortzaileak bestearen alderantzizkoa burutu behar duenez, G_{I2T} sareak kodetu duen irudia berreskuratu behar du. Hortaz, bigarren sortzaile honek $G_{T2I} : \mathbb{R}^{1536} \longrightarrow \mathbb{R}^{227\times227\times3}$ funtzioa burutzen du.

Geruza	**Input Size**	**Output Size**	**F. Size**	**F. Count**	**Stride**	**Padding**	φ
1. FC	1536	4096					ReLU
2. FC	4096	4096					ReLU
3. FC	4096	6x6x256					ReLU
4. Conv^T	6x6x256	13x13x384	3	384	2	0	ReLU
5. Conv^T	13x13x384	27x27x384	3	384	2	0	ReLU
6. Conv^T	27x27x384	55x55x256	3	256	2	0	ReLU
7. Conv^T	55x55x256	55x55x96	5	96	1	2	ReLU
8. Conv^T	55x55x96	227x227x3	11	3	4	0	Sigmoid

4.2 Taula: G_{T2I} sortzailea osatzen duten geruzak. Zutabe bakoitzaren esanahia 4.1 taulan aurki daiteke. *3. FC* geruza-dentsoaren bukaeran, 9216 elementuko bektorea matrize hirudimentsional batera bihurtzen da, konboluzio-geruzak aplikatu ahal izateko.

G_{I2T} sarean ez bezala, G_{T2I} sareak Conv$^{\mathrm{T}}$ *konboluzio irauliak* erabiltzen ditu *upsampling* eragiketa burutzeko, hots, irudien tamaina handitzeko. Funtsean, konboluzio iraulietan pixel bakoitza matrize edo filtro bakoitzarekin bidertzen da, filtroaren tamaina bereko pixel-multzoak hartu beharrean. Ondoren, biderketa guztien emaitza batzen da, irudi handiago batekin amaituz —4.4 irudia—.



4.4 Irudia: Konboluzio irauliaren adibidea, non *I* irudiari *W* konboluzio irualia aplikatzen zaion *O* irudia lortzeko.

Gainera, G_{T2I} sareek ez dute *max-pooling* eragiketaren alderantzizkoa burutzen. Izan ere, *pooling* eragiketek ez dituzte alderantzizkorik, pixel-multzo baten batez-besteko edo maximoak izanik ezin baitira hasierako pixelak berreskuratu. G_{T2I} sortzaileak irudiak itzuli behar dituenez, eta irudi horien pixelen balioak $[0,1]$ tartean definituta daudenez, *Sigmoid* funtzioa erabili da azken geruzako aktibazio-funtzio bezala.

Diskriminatzaileak, berriz, ez dira sortzaileak baino konplexuagoak. D_I sareak, adibidez, $D_I : \mathbb{R}^{227\times227\times3} \longrightarrow [0,1]$ funtzioa burutzen du, non 0 batek sarrerako irudia G_{I2T} sareak sortu duela esan nahi duen; eta batekoak, aldiz, irudi orijinal bat dela. G_{I2T} sarearen arkitektura aprobetxatuz, honen azken geruza bakarrik aldatu da D_I diskriminatzailea eraikitzeko. Beraz, diskriminatzaile honek 4.1 taulako arkitektura jarraitzen du, azken geruza 4.3 taulan azalduz.

Geruza	**Input Size**	**Output Size**	φ
8. FC	4096	1	Sigmoid

4.3 Taula: D_I diskriminatzailearen azken geruza-dentsoa. Diskriminatzaile honen beste zazpi geruzek G_{I2T} sarearen egitura bera dute.

Azken diskriminatzailearekin bukatzeko, aipatu diren lau sareetatik bakarra da irudiak manipulatzen edo sortzen ez dituena. Horregatik, D_I diskriminatzailea neurona-sare sinpleena da. Hitz-gutxitan, hiru geruzako pertzeptroia da, $D_T : \mathbb{R}^{512} \longrightarrow [0,1]$ funtzioa betetzen duena, irudiak diskriminatu ordez, **t** goiburuko-bektoreak aztertzen dituelarik. Bere egitura 4.4 taulan azaltzen da.

Geruza	**Input Size**	**Output Size**	φ
1. FC	512	1024	ReLU
2. FC	1024	1024	ReLU
3. FC	1024	1	Sigmoid

4.4 Taula: D_T diskriminatzailearen geruzen egitura.

4.3 Ikasketa-prozesua

Burutu behar den mapaketa egin behar duten sareak definitu eta hauen arteko elkareragina aztertu badira ere, sare hauetan agertzen diren pisuak nolabait ikasi behar dira. Horregatik, azpiatal honetan pisu horien ikasketa-prozesua azalduko da.

Ikasketa-prozesua bi fase ezberdinetan banatu da. Izan ere, *vSTS* datu multzoan eskura dauden instantzia kopurua ez da nahikoa *DiscoGAN-M*^{3} sistemaren modalitate anitze-

ko mapaketak ikasteko, sistemak erabilitako datu multzoetako instantzietatik orokortzeko ahalmena lortzea nahi da eta —eta azkeneko honetarako ezinbestekoa da datu multzo handiak eskura edukitzea—.

Lehenengo fasean, hortaz, sistemak orokortzeko ahalmena lortzea da helburu nagusia; eta bigarrenean, aldiz, sistema hau *vSTS* atazan berfintzea.

4.3.1 Lehen fasea

Esan bezala, fase honetan *DiscoGAN-M*^{3} sistemaren modalitate anitzeko mapaketak ikasteko ahalmena lortzea da helburua. Horretarako, *vSTS* datu multzoa baino askoz handiagoak diren datu multzo bat behar da; eta kasu honetan *MS-Coco* datu multzoaren azpiatal bat erabili da.

Gaur egun *MS-Coco* ia 330K irudiz eta hauen goiburukoez osatuta dago, eta, berez, arkitektura ahal den hoberen entrenatzekoan irudi guzti horiek erabili beharko lirateke —datu multzo handiago batek orokortzeko gaitasun handiagoa ematen dio eta—. Hala ere, irudi kopuru hori oso handia da etxeko ordenagailuan maneiatzeko. Horregatik, 82783 irudi bakarrik erabili dira, ia 14GB-ko memoria okupatzen dituztenak. Irudi hauek 2014 urtean *Train* azpimultzora gehitu zirenak dira.

Irudi bakoitzak bost goiburuko dituenez, lehenengo fasean erabiltzeko guztira 413915 irudi eta goiburuko pare ezberdin daude eskura. Datu kopuru hauek fase honen helburua betetzeko nahikoak izango direla suposatu da.

Behin erabiliko den datu multzoa definituta dagoela, ikasketa-prozesua nola burutu den azalduko da. Funtsean, prozesuko iterazio, belaunaldi edo *epoch* bakoitzean —prozesu iteratibo bat baita—, hainbat *mini-batch* aukeratzen dira. *Mini-batch* bakoitzeko aurreranzko propagazio bat burutzen da sisteman zehar, galera-funtzio ezberdinak kalkulatuz. Ondoren, galera-funtzioek itzuli dituzten balioen arabera, atzeranzko propagazioa burutzen da, sistemaren pisuak eguneratuz *mini-batch* bakoitzean.

Erabili den txartel-grafikoaren 6GB-ko memoria dela eta, *mini-batch* bakoitzean 20 $\mathbf{x}_I$ irudi eta $\mathbf{t}_T$ goiburuko pare bakarrik sartu dira —memoria kapazitatea ez gainditzeko—. Irudiak ausaz aukeratzen dira, eta behin irudiak aukeratuta izanik, hauen goiburukoak ausaz hartzen dira ere bai.

$\mathbf{f}_T$ bektoreak ezagunak ez direnez, aurreko *mini-batch*arekin lortu den $\mathbf{f}_{IT}$ bektoreak era-

bili dira. $\mathbf{f}_T^{(i)}$ ikasketa-prozesu osoko i. *mini-batch*ean erabiltzen diren bektoreak badira, 4.7 ekuazioek bektore hauek nola aukeratzen diren definitzen dute.

$$\mathbf{f}_T^{(0)} = \mathbf{f}_{IT}^{(0)} \tag{4.7a}$$

$$\mathbf{f}_T^{(i)} = \mathbf{f}_{IT}^{(i-1)} \quad \text{for } i > 0 \tag{4.7b}$$

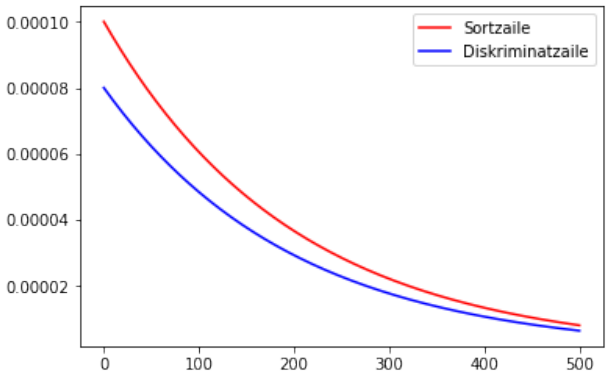
Irudi bakoitza behin bakarrik aukeratzen da *epoch* bakoitzean, eta *epoch* bakoitza 1241 *mini-batch* ezberdinez osatzen da —belaunaldi bakoitzean irudi guztietatik ausazko %30 irudi edo %6 irudi/goiburuko pare aukeratuz—. Belaunaldi oso bat prozesatzeko 15 minutu inguru behar direnez, prozesu osoa 500 belaunaldietara mugatu da.

$$\mu_G = 10^{-4} \cdot 0.995^{\text{epoch}} \tag{4.8a}$$

$$\mu_D = 8 \cdot 10^{-5} \cdot 0.995^{\text{epoch}} = 0.8 \cdot \mu_G \tag{4.8b}$$

Ikasketa burutzeko, Adam optimizazio algoritmoa erabili da [44], oinarrizko *SGD* algoritmoaren antzekoa dena, baina sistemako parametro bakoitzak ikasketa-tasa pertsonal bat du —parametro bakoitzak emaitzetan duen eraginaren menpekoa dena—.

Gainera, ikasketa-tasa ezberdinak erabili dira sortzaile eta diskriminatzailetan —4.8 ekuazioak jarraituz, hurrenez hurren—. 4.5 irudian 500 belaunaldietan zehar burutzen duen eboluzioa antzematen da, baina kontuan izan behar da parametro bakoitzak uneko ikasketa-tasaren bariazio bat erabiliko duela; hots, grafikoan azaltzen denaren antzekoa.



4.5 Irudia: Ikasketa-indizeen balio orokorrak ikasketa-prozesuko lehen fase osoan zehar.

Azken finean ikasketa-tasa baxuago batek ikasketa-prozesua mantsotzen du, eta diskriminatzaileen ataza sortzaileena baino errazagoa denez, bien arteko oreka mantentzen saiatzeko hautaketa hori burutu da. Gainera, *mini-batch* bakoitzean sortzaile edota diskriminatzaileei bakarrik aplikatzen zaie atzeranzko propagazioa, ez biei. 10 *mini-batch* ezberdinetik bat diskriminatzaileen pisuak eguneratzeko erabiltzen da, eta beste bederatziak sortzaileena eguneratzeko. Aukera hauek guztiak sortzaileak eta diskriminatzaileen ikasketak erritmo berean aurrera eramateko egin dira.

$$L_{\text{Total}\cdot \text{G}_{I2T}} = L_{\text{G}_{I2T}} \cdot (1-r) + \frac{L_{\text{CONST}_I} + L_{\text{CONST}_T} + L_{\text{CONST}_S}}{3} \cdot r \tag{4.9a}$$

$$L_{\text{Total}\cdot \text{G}_{T2I}} = L_{\text{G}_{I2T}} \cdot (1-r) + \frac{L_{\text{CONST}_I} + L_{\text{CONST}_T}}{2} \cdot r \tag{4.9b}$$

Berreraikitze eta sortzaileen galera-funtzioak sortzaileen pisuak eguneratzeko erabiltzen direnez, bien batuketa haztatua erabili da sortzaileen galera-funtzio totala bezala —4.9 ekuazioa—. Batuketa haztatu horren pisuak fasean zehar aldatu dira, lehenengo belaunaldietan $L_{\text{G}_{I2T}}$ eta $L_{\text{G}_{T2I}}$ balioei garrantzi gehiago emateko. Lehenengo 60 belaunaldietan, $r = 0.02$ erabili da; eta, ondoren, $r = 0.5$.

4.3.2 Bigarren fasea

Behin modalitate anitzen arteko mapaketak ikasi dituela suposatuz, *vSTS* atazan afinatuko da sistema. Bigarren fase honetan irudietatik goiburukoak lortzen dituen G_{I2T} sortzailea bakarrik erabiliko da. Horrela, irudien $\mathbf{t}_{IT}$ goiburuko sortuak eta $\mathbf{t}_T$ goiburuko orijinalak erabiliz, antzekotasun semantikoak aztertzeko bikote gehiago lor daitezke. Bigarren fasean sistemak lortuko duen orokortze ahalmenari garrantzia eman zaio.

Demagun *vSTS* datu multzoko instantzia bat eskura dagoela, $\mathbf{x}_I^A$ eta $\mathbf{x}_I^B$ irudi, $\mathbf{t}_T^A$ eta $\mathbf{t}_T^B$ goiburuko, eta s antzekotasun semantikoaren balioaz osatuta. $STS : \mathbb{R}^{512+512} \longrightarrow [0,5]$ funtzioak bi esaldiren arteko antzekotasun semantikoa kalkulatzen badu, 4.10 ekuazioak bete beharko lirateke.

$$STS(\mathbf{t}_T^A, \mathbf{t}_T^B) = s \tag{4.10a}$$

$$STS(\mathbf{t}_T^A, \mathbf{t}_{IT}^B) = s \tag{4.10b}$$

$$STS(\mathbf{t}_{IT}^A, \mathbf{t}_T^B) = s \tag{4.10c}$$

$$STS(\mathbf{t}_{IT}^A, \mathbf{t}_{IT}^B) = s \tag{4.10d}$$

Beraz, ikasitako G_{I2T} sortzailea erabiltzeak eskura dauden instantzia kopurua lau aldiz handitzea dakar. Izan ere, hasiera batean $\mathbf{t}_I^A$ eta $\mathbf{t}_I^B$ bektoreak bakarrik daude eskura, eta, goiburukoen $\mathbf{x}_I^A$ eta $\mathbf{x}_I^B$ irudiak eta G_{I2T} sarearen bitartez, $\mathbf{t}_{IT}^A$ eta $\mathbf{t}_{IT}^B$ bektoreak lortu daitezke —lau bektore horiekin bi esaldien goiburukoen arteko lau konbinazio lortzen direlarik—.

Modu honetan modalitate anitzeko mapaketa erabiliz, eskura dauden instantzia kopurua erraz handitu daiteke. Lehen definitu den *STS* funtzioa *Multilayer-Perceptron* edo *MLP* bat erabiliz burutuko da, 4.5 taulan honen geruzak definitzen direlarik.

Geruza	**Input Size**	**Output Size**	φ
1. FC	1024	1024	ReLU
2. FC	1024	1024	ReLU
3. FC	1024	1	Sigmoid

4.5 Taula: *STS* burutzen duen sarea. Azkeneko geruzan *Sigmoid* funtzioa aplikatu ondoren lortutako emaitzak eskalatu egiten dira, $[0,1] \longrightarrow [0,5]$ tartera joateko.

$$L_A = \text{MSE}(\mathbf{t}_T^A, \mathbf{t}_T^B), s) \tag{4.11a}$$

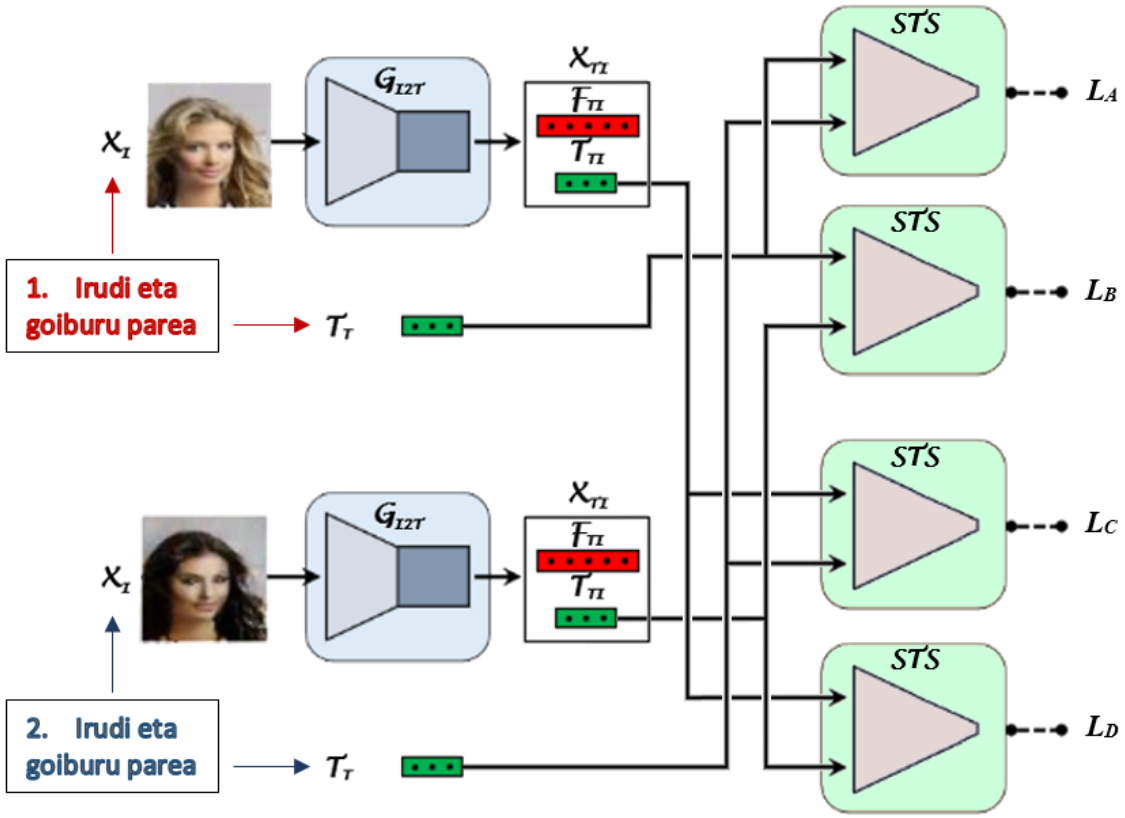
$$L_B = \text{MSE}(\mathbf{t}_T^A, \mathbf{t}_{IT}^B), s) \tag{4.11b}$$

$$L_C = \text{MSE}(\mathbf{t}_{IT}^A, \mathbf{t}_T^B), s) \tag{4.11c}$$

$$L_D = \text{MSE}(\mathbf{t}_{IT}^A, \mathbf{t}_{IT}^B), s) \tag{4.11d}$$

Bigarren fase honen arkitektura 4.6 irudian irudikatzen da. Ikus daitekeenez, lau galera-balio lortzen dira, balio bat goiburuko konbinazio bakoitzeko: L_A, L_B, L_C eta L_D —4.11 ekuazioan definituta, hurrenez hurren—. Lau galera balio horiek fase honetako L_{Total} galera-funtzioan erabiltzen dira, aurreko lau balioen batez-bestekoa kalkulatzen duena —4.12 ekuazioa—, eta fase honetako ikasketa-prozesuan minimizatu nahi den balioa.

$$L_{\text{Total}} = \frac{L_A + L_B + L_C + L_D}{4} \tag{4.12}$$

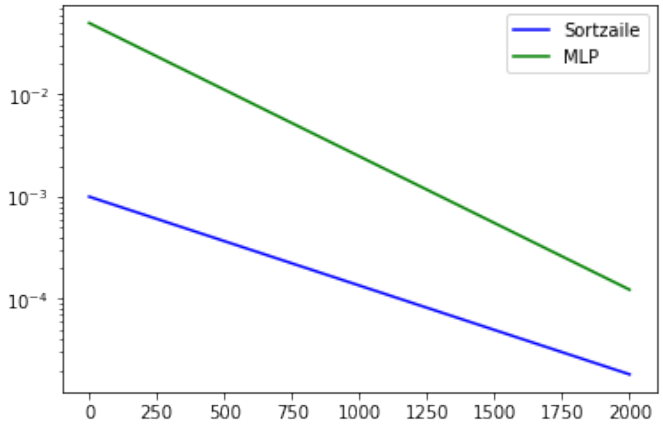


4.6 Irudia: DiscoGan-M^3 sistemaren azpiatal bat erabiliz *vSTS* sistema ebazten duen arkitektura, bigarren ikasketa-fasean ikasten dena.

Bigarren fase hau gainbegiratua da —aurrekoa ez bezala—. *vSTS* datu multzoa erabili da ikasketa-fase honetarako, *Train*, *Dev* eta *Test* multzoak sistema ikasteko, sistemaren ikasketa aztertzeko, eta lortutako sistemaren errendimendua ikusteko erabiltzen direlarik, hurrenez hurren. Sortzaile bat eta *MLP* bat erabiltzen direnez, *mini-batch* handiagoak erabili dira, *mini-batch*aren instantzia bakoitza bi irudi eta bere goiburukoez osatzen bada ere. Guztira, *mini-batch* bakoitzean 64 instantzia sartu dira, eta 2000 belaunalditan zehar garatu da ikasketa.

Aurreko fasean ez bezala, *Dev* azpimultzoa hoberen ebazten duen belaunaldia izango da aukeratuko dena —ez azkenekoa—, orokortze hoberena burutzen duena hain zuzen ere.

SGD optimizazio algoritmoa erabili da, *Adam* algoritmoak baino orokortze hobeagoak egiten baititu [45], orokorrean *Adam*ek azkarrago konbergitzen badu ere. Sortzailearen kasuan $1 \cdot 10^{-3}$-ko ikasketa-tasa erabili da, iterazio bakoitzean 0.998 konstantearekin bi-



4.7 Irudia: Ikasketa-indizeen balioak ikasketa-prozesuko bigarren fase osoan zehar —ardatz horizontala belaunaldia, eta bertikala ikasketa-tasaren balioa izanik—.

dertuz. *MLP*-ren kasuan, aldiz, $5 \cdot 10^{-2}$-ko ikasketa-tasa, 0.997-ko konstantearekin bidertuz *epoch* bakoitzean —4.7 irudia—.

Azkenik, bi faseetako ikasketa-prozesuan sareetako geruza-dentsoetan *Dropout* teknika erabili da, $p = 0.5$ erabiliz. Honek, geruza horietako neurona erdiak baliogabetzen ditu, neurona horiek ausaz aukeratzen direlarik *mini-batch* bakoitzean. Teknika hau, sareen orokortze-ahalmena handitzeko erabiltzen da.

5. KAPITULUA

Esperimentuak

Hainbat esperimentu ezberdin burutu dira *STS* eta *vSTS* atazetan, bai ez-gainbegiratuak eta baita gainbegiratuak ere. Esperimentu hauen helburu nagusia 2. kapituluan azaldu diren sistemen errendimendua *STS-B*, *vSTS v1.0* eta *vSTS v2.0* datu multzoetan aztertzea da, ea modalitate anitzak erabiliz emaitzak hobetzen diren ikusiz. Gainera, ondorengo puntuak aztertu nahi dira:

- Testuen, irudien eta modalitate anitzeko errepresentazioen arteko errendimendua ikustea.
- Esperimentuetan erabilitako datu multzo ezberdinen zailtasuna antzematea, eredu ezberdinen errendimendua aztertuz multzo bakoitzean.
- *DiscoGAN-M*^{3} arkitekturaren ikasketa eta errendimendua analizatu eta beste ereduekin konparatzea.

vSTS v2.0 datu multzoan modalitate bakar eta anitzeko esperimentuak egin daitezke, noski, baina hasiera batean datu multzo hori eskura ez zegoenez, *vSTS v1.0* datu multzoa proba ez-gainbegiratuetan bakarrik erabili zen, modalitate anitzeko sistemen ikasketa-prozesurako ez baitzeuden nahiko instantzia. Gainera, kontuan izan behar da *STS-B* esaldien arteko antzekotasun semantikoa aztertzeko balio duela bakarrik —irudirik ez ditu eta—.

Esperimentuetan zehar datu multzo hauen bi bariazio agertzen dira. Izan ere, sistema batzuen ikasketa-prozesuan, hau da, *Transfer-learning* burutu baino lehenagoko ikasketa-

prozesuetan, hainbat datu multzo ezberdinetatik lortutako instantziez baliatu dira. Proiektuko hainbat esperimentu ez-gainbegiratu eta gainbegiratuetako sistemen ikasketa-prozesuan erabili diren instantziak aipatutako datu multzoetatik kendu nahi izan dira.

1. ***vSTS* gabeko *STS-B***: izenak esan bezala *STS-B* datu multzoari goiburukoen azpimultzoko instantzia erdiak kendu zaizkio bariazio honetan, *vSTS v1.0* datu multzoan azaltzen diren instantziak kendu direlarik, hain zuzen ere. 5.1 taulan datu multzoaren bariazio honen instantzien distribuzioa azaltzen da, eta, guztira, 1500 instantzia kendu zaizkio —2014 eta 2015 urteetan lortutako *Caption* azpimultzoko instantziak direnak—. Kapitulu honetako tauletan, datu multzoa *STS-B** bezala adieraziko da.

	Train	**Dev**	**Test**	**Guztira**	**Train**	**Dev**	**Test**	**Guztira**
Berriak	3299	500	500	4299	%46.29	%7.01	%7.01	%60.31
Goiburuko.	1000	375	375	1750	%14.03	%5.26	%5.26	%24.55
Foroak	450	375	254	1079	%6.31	%5.26	%3.57	%15.14
Guztira	4749	1250	1129	7128	%66.63	%17.53	%15.84	%100.0

5.1 Taula: Taula hauetan azpimultzo bakoitzak duen instantzia kopuru eta proportzioak aurki daitezke, hurrenez hurren. Instantzia bakoitzak bi esaldi ditu; beraz, esaldi kopurua tauletako balioen bikoitza da.

2. ***MS-Coco* gabeko *vSTS v2.0***: Aurreko kasuan bezala, *vSTS v2.0* datu multzoan erabilitako instantzia batzuk kendu dira, hau da, *MS-Coco* datu multzoak agertzen zirenak kendu dira.

	Train	**Dev**	**Test**	**Guztira**	**Train**	**Dev**	**Test**	**Guztira**
Flickr30k	221	114	104	439	%17.42	%8.99	%8.2	%34.61
vSTS v1.0	400	208	221	829	%31.55	%16.4	%17.43	%65.39
Guztira	621	322	325	1268	%48.97	%25.39	%25.63	%100.0

5.2 Taula: Taula hauetan azpimultzo bakoitzak duen instantzia kopuru eta proportzioak aurki daitezke, hurrenez hurren.

Datu multzo hau nahiko txikia da, *vSTS v2.0* datu multzoaren %47.37 instantzia izanik. Hori dela eta, sare ezberdinen ikasketarako txikiegia denez, esperimentu ez-gainbegiratuetan erabili da bakarrik. Kapitulu honetako tauletan datu multzoa *vSTS v2.0** bezala adieraziko da.

5.1 Ebaluazio metrikak

Esperimentuei buruz hitz egin baino lehen, sistemen errendimendua ebaluatzeko aztertu diren metrika edo irizpide ezberdinak komentatutako dira.

- *Pearson* korrelazioa: bi aldagai kuantitatiboren arteko korrelazio linealaren sendotasuna neurtzen du. Ondorengo kasuetan aldagai horiek sarearen $\mathbf{x}$ estimazioak eta $\mathbf{y}$ antzekotasun semantikoaren balio errealak izango dira. Bi aldagai horien arteko korrelazio honen kalkulua 5.1 ekuazioaren bidez lortzen da, aldagaien arteko kobariantza estandarizatua erabiliz —3.1 eta 3.2 ekuazioetan aipatua—.

$$\rho_{p_{\mathbf{x},\mathbf{y}}} = \frac{\text{cov}(\mathbf{x},\mathbf{y})}{\sigma_{\mathbf{x}}\sigma_{\mathbf{y}}} = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{\mathbf{x}_i-\overline{\mathbf{x}}}{\sigma_{\mathbf{x}}}\right)\left(\frac{\mathbf{y}_i-\overline{\mathbf{y}}}{\sigma_{\mathbf{y}}}\right) \tag{5.1}$$

  Bi aldagaien korrelazioaren $\rho_{p_{\mathbf{x},\mathbf{y}}}$ koefizientea $[-1,1]$ tartean definituta dago. Balio hau zerora hurbiltzen denean korrelazio lineal oso baxua daukatela esan daiteke; eta zerotik urruntzen doan heinean, korrelazio hori sendoagoa bihurtzen da. Zeinu positibo ala negatiboaren arabera, $\mathbf{x}_i$ aldagai baten balioa handitzean, $\mathbf{y}_i$ bestearena handitzen ala txikitzen da, hurrenez hurren.

- *Spearman* korrelazioa: *Pearson*en korrelazioa bezala, bi aldagai kuantitatiboren arteko korrelazioaren sendotasuna neurtzen du, baina korrelazio horrek ez du zertan lineala izan behar. Kobariantza estandarizatuan oinarritu beharrean, aldagai bakoitzeko elementuak ordenatzen dira. Orden horiek kontuan edukiz, $R_{\mathbf{x_i}}$ eta $R_{\mathbf{y_i}}$ balioek i. elementuaren posizioa adierazten dute ordena horretan. Beraz, 5.2 ekuazioak $\rho_{s_{\mathbf{x},\mathbf{y}}}$ korrelazioa itzultzen du, non $d_i = (R_{\mathbf{x}_i} - R_{\mathbf{y}_i})^2$

$$\rho_{s_{\mathbf{x},\mathbf{y}}} = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)} \tag{5.2}$$

  Kasu honetan ere, bi aldagaien korrelazioaren $\rho_{\mathbf{x},\mathbf{y}}$ koefizientea $[-1,1]$ tartean definituta dago, balio horiek modu berean interpretatzen direlarik.

- Bataz-besteko errore koadratikoa: *MSE* edo *Mean Square Error* erabiliz, sareak hainbat instantzietan izan duen errorea kalkulatu daiteke —2.3 ekuazioan agertzen da—. Irizpide hau hainbat galera-funtzioetan erabili da.

Funtzio honen balioak $[0,\inf)$ tartean egon daitezke, gero eta balio txikiago batek batez-besteko errore txikiago bat lortu dela esan nahi duelarik. *STS* eta *vSTS* atazen kasuan balioak $[0,5]$ tartean daudenez, lortu daiteken erroreak $[0,25]$ tartean aurkituko dira.

Spearman korrelazioak kasu batzuetan kobariantzan oinarritutako korrelazioak baino errendimendu hobea ematen badu ere [46], *Pearson* korrelazioa aukeratu da sare ezberdinen errendimendua aztertzeko irizpide bezala —gehien erabiltzen dena izan ohi da eta—.

MSE erabiltzeak zentzua badu ere, problema hauetarako ez da irizpide zuzena bezala kontsideratzen, sare hauen errendimendua aztertzean korrelazioari garrantzi handiagoa ematen zaio eta. Gainera, bi esaldien arteko antzekotasuna neurtzeko kosinuaren-antzekotasuna erabiltzen bada, adibidez, *MSE* irizpideak emaitza okerragoak ematen ditu, estimazioak eta balio errealak tarte ezberdinetan definituta daudelako —$[0,1]$ eta $[0,5]$, hurrenez hurren—. Estimazioak 5 aldiz handitzen badira ere, bere errendimendua ez da ona. Izan ere, *MSE* irizpidea erabiltzean, korrelazio linealak detektatzeko $y = x$ ekuazioa jarraitu behar du; baina, beste bi irizpideen kasuan, $y = m \cdot x$ izan daiteke, $m > 0$ izanik —ebaluazio-irizpideari askatasun-gradu bat emanik—.

5.2 Esperimentu gainbegiratugabeak

5.2.1 Ereduak

Esperimentu gainbegiratuetan *Transfer-learning* teknika aplikatu da, hots, eredu ezberdinek *STS* edo *vSTS* atazak ez diren beste ataza bat ikastea eta ikasitakoa ataza horietan aplikatzea. Sistemaren arabera esaldiaren errepresentazioak modu ezberdinean lortu dira, baina behin esaldi-bektoreak kalkulatuta daudela, bi bektoreen arteko kosinuaren-antzekotasuna kalkulatzen da, esaldien antzekotasuna $[0,1]$ tartean kalkulatzeko —5.3 ekuazioan ikus daiteke bere definizioa—. *Pearson* korrelazioaren bidez, lortutako balioak ebaluatzen dira, eta, hortaz, sistemen errendimendua ere bai.

$$\text{sim} = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{5.3}$$

Proba ez-gainbegiratuetan 4 sistema ezberdin erabili dira esaldi-bektoreak lortzeko:

- *GloVe*: esaldi bakoitzean agertzen diren hitzen *GloVe* hitz-bektoreen zentroidea edo batez-bestekoa bi modutan kalkulatu da, hitz hutsak kontuan hartuta edo alde batera utzita.

- *BERT*: *BERT Transformer*ra eta honen tokenizazio berezia erabiliz, esaldi bakoitza bere kabuz prozesatu da. *Transformer*raren azken geruzako token guztiak erabiliz, hauen zentroidea kalkulatu da, hitz hutsen tokenak kontuan hartuta edo alde batera utzita —beti ere [CLS] eta [SEP] tokenak kontuan hartu gabe—. Gainera, arkitektura honek ikasteko baliatzen den *NSP* ataza aprobetxatuz, ataza hori ikasteko erabiltzen den *CLS* geruza erabili da beste esaldiaren errepresentazio modura, esaldi-bektoreak hiru modu ezberdinetan kalkulatuz *Transformer* honekin.

- *GPT-2*: aurreko *Transformer*ran bezala, azkeneko geruzako token ezberdinen zentroidea erabili da, hitz hutsen tokenak kontuan hartuta edo alde batera utzita. Beti ere, [CLS] eta [SEP] tokenak ez dira kontuan hartu zentroideak sortzerakoan.

- *VSE++*: Azkeneko sistema hau modalitate anitzeko errepresentazioak erabiltzen dituenez, irudiak eta testua bateratu daitezke. Beraz, irudiak eta hitzak espazio bereko bektoretan erabili daitezkeenez, irudi pareen arteko antzekotasuna, testu pareen artekoa eta bien arteko batez-bestekoak erabili dira.

5.2.2 Emaitzak

Ondorengo tauletan esperimentu ez-gainbegiratuetan lortutako korrelazioak azaltzen dira. Korrelazio bakoitza kolore batez lagunduta dago. Kolore hori berdea bada, korrelazio altua lortu dela esan nahi du. Kolorea gorria bada, aldiz, korrelazio nahiko baxua lortu dela esan nahi du. Beste aldetik, taula urdinetan *STS* ataza burutu da, testua bakarrik kontsideratu delarik; eta, kolore arrosako tauletan, testuak eta irudiak erabili dira.

5.3, 5.4 eta 5.5 tauletan modalitate bakarra erabiltzen dituzten sistemen korrelazioak azaltzen dira, taula bakoitzean datu multzo ezberdin baten emaitzak agertzen direlarik. 5.6, 5.7 eta 5.8 tauletan, aldiz, *VSE++* sistemaren bi aldaeren emaitzak aurki daitezke.

		vSTS v1.0
BERT	All Tokens	54,5446%
	Remove SW	70,6316%
	CLS layer	12,1673%
GloVe	All Tokens	65,2677%
	Remove SW	72,1693%
GPT-2	All Tokens	32,3396%
	Remove SW	34,3349%

5.3 Taula: *vSTS v1.0* datu multzoko esperimentu ez-gainbegiratuetan lortu diren korrelazioak, modalitate bakarreko sistemak erabiliz. Sistema bakoitzean gehienez hiru modu ezberdinetan lortu dira esaldien errepresentazioak. *All tokens*: token guztiak erabiltzen dira —[CLS] eta [SEP] kontuan hartu gabe—. *Remove SW*: hitz hutsak ez dira kontuan hartzen. *CLS layer*: CLS geruzako errepresentazioa erabiltzen da.

		vSTS v2.0			
		Train	Dev	Test	All
BERT	All Tokens	51,0302%	44,2468%	47,4643%	48,3077%
	Remove SW	64,1224%	59,3042%	61,2003%	62,1634%
	CLS layer	5,9543%	4,4789%	8,5952%	6,1205%
GloVe	All Tokens	50,1314%	49,7314%	51,1214%	50,2121%
	Remove SW	57,5919%	57,9732%	58,6719%	57,9108%
GPT-2	All Tokens	31,2454%	16,5023%	18,5158%	20,3237%
	Remove SW	31,4105%	19,0572%	19,9257%	21,8089%

5.4 Taula: *vSTS v2.0* datu multzoko esperimentu ez-gainbegiratuetan lortu diren korrelazioak, modalitate bakarreko sistemak erabiliz.

		STS-B*			
		Train	Dev	Test	All
BERT	All Tokens	51,3751%	59,9456%	47,0314%	52,9916%
	Remove SW	63,4909%	71,0815%	51,3489%	62,5781%
	CLS layer	0,5441%	-2,3846%	0,9443%	0,3700%
GloVe	All Tokens	43,2596%	49,8475%	33,6014%	41,5411%
	Remove SW	56,3617%	65,6165%	48,6071%	56,8792%
GPT-2	All Tokens	19,7835%	24,1111%	20,9841%	21,1348%
	Remove SW	20,7755%	22,2117%	19,0762%	20,6116%

5.5 Taula: *STS-B** datu multzoko esperimentu ez-gainbegiratuetan lortu diren korrelazioak, modalitate bakarreko sistemak erabiliz.

		vSTS v1.0		
		Image	Sentence	Average
VSE++	VGG19	67,7517%	89,0152%	86,6065%
	ResNet152	72,1393%	88,8498%	87,5305%

5.6 Taula: *vSTS v1.0* datu multzoko esperimentu ez-gainbegiratuetan lortu diren korrelazioak, modalitate anitzeko sistemak erabiliz —bi konboluzio-sare ezberdin erabiliz—.

		vSTS v2.0					
		Train			Dev		
		Image	Sentence	Average	Image	Sentence	Average
VSE++	V.	64,7792%	83,6208%	82,3730%	64,0747%	82,3175%	81,9044%
	R.	69,4699%	82,7132%	83,5732%	67,5698%	81,6196%	82,5397%
		Test			All		
		Image	Sentence	Average	Image	Sentence	Average
VSE++	V.	64,2743%	82,5625%	81,5563%	64,5132%	83,0469%	82,0714%
	R.	67,8623%	81,2731%	81,6213%	68,6172%	82,0976%	82,8447%

5.7 Taula: *vSTS v2.0* datu multzoko esperimentu ez-gainbegiratuetan lortu diren korrelazioak, modalitate anitzeko sistemak erabiliz.

		vSTS v2.0*					
		Train			Dev		
		Image	Sentence	Average	Image	Sentence	Average
VSE++	V.	64,8527%	86,1587%	84,5892%	58,9147%	83,8359%	81,7877%
	R.	69,3076%	85,7721%	85,3997%	64,6932%	84,8180%	83,9659%
		Test			All		
		Image	Sentence	Average	Image	Sentence	Average
VSE++	V.	63,8133%	84,5368%	83,2187%	63,2703%	85,1783%	83,5888%
	R.	67,8623%	84,0294%	83,4618%	67,7879%	85,1159%	84,5854%

5.8 Taula: *vSTS v2.0** datu multzoko esperimentu ez-gainbegiratuetan lortu diren korrelazioak, modalitate anitzeko sistemak erabiliz.

5.3 Esperimentu erdi-gainbegiratuak

USE sistemak hainbat ataza ikusi ditu bere ikasketa-prozesuan zehar, horietako bat *STS* ataza izanik. Gainera, *STS-B* datu multzoa erabili dute ataza hau ikasteko. Hortaz, guztiz gainbegiratua ez bada ere, *USE* sistema *STS* eta *vSTS* atazak ebazteko orduan erdi-gainbegiratua dela esan daiteke.

USE sistemak esaldi-bektore bat itzultzen du zuzenean. Hortaz, esaldi-bektoreen kosinuaren-antzekotasuna eta *Pearson* korrelazioak erabili dira sistema ebaluatzeko, aurreko kasuetan bezala. Emaitzak 5.9 taulan aurki daitezke.

	Train	Dev	Test	All
vSTS v2.0	74,7907%	75,8676%	73,6657%	74,7453%
STS-B*	68,5219%	75,0637%	70,2085%	70,3134%

5.9 Taula: Esperimentu erdi-gainbegiratuetan lortu diren korrelazioak, *USE* sistema erabiliz.

5.4 Esperimentu gainbegiratuak

5.4.1 Ereduak

Bi sistema ezberdin erabili dira esperimentu gainbegiratuetan, *BERT Transformer*ra eta *DiscoGAN-M*^{3} arkitekturak erabiliz.

*Transformer*raren kasuan, bi bariazio erabili dira. Lehenengoan azken geruzako [CLS] tokenaren gainean erregresio-geruza bat gehitu da, *Sigmoid* aktibazio-funtzioarekin eta *MSE* galera-funtzioa erabiliz. Ondoren, emaitza horiek $[0,5]$ tartera eskalatzen dira, zuzenean antzekotasun semantikoa kalkulatzeko —kosinuaren-antzekotasuna erabili gabe—.

Bigarrenean, aldiz, erregresio-geruza bat erabili beharrean sei neuronako geruza-dentso bat erabili da, geruza horri *Softmax* funtzioa aplikatuz. Horrela, neurona horien balioa guztira $\sum_{i=0}^{5} x_i = 1$ izango da, eta, 5.4 ekuazioa erabiliz, $[0,5]$ tarteko s antzekotasuna lortu daiteke.

$$s = \sum_{i=0}^{5} i \cdot x_i \tag{5.4}$$

Ikasketa-prozesuan zehar *STS-B* edo *STS-B** erabili bada, 25 belaunaldiz burutu da berfintzea; bestela, 125 belaunaldiz —belaunaldi bakoitzean *Train* eta *Dev* azpimultzoko instantzia guztiak erabiltzen baitira, eta *STS-B* datu multzoa besteak baino zerbait handiagoa baita—. Belaunaldi guztietatik, *Dev* azpimultzoan galera-balio txikiena duena aukeratzen da, *overfitting* ekiditeko asmoarekin, eta gehitzen zaion azken kapari *Dropout*-a aplikatzen zaio $p = 0.5$ izanik.

Ikasketa-indizearen aldetik, $\eta = 10^{-4}$ bezala hasieratzen da, bost belaunaldiro 0.85 konstantearekin bidertuz. Optimizazio-algoritmoa bezala *SGD* erabili da.

DiscoGAN-M^3 arkitekturan, aldiz, 4. kapituluan aipatutako ikasketa-prozesua jarraitu da. Hala ere, bigarren fasean hainbat aldaera ezberdin erabili dira esaldien arteko antzekotasun semantikoak neurtzeko garaian.

Aldaera hauetan kosinu-antzekotasuna edo *Multilayer-Perceptron* bat erabili da bi esaldi-bektoreen arteko antzekotasuna estimatzeko. *MLP* sarearen ikasketa hiru modu ezberdinetan burutu da. Lehenengoan *USE* sistemaren esaldi-bektoreak erabili dira, bigarrenean *DiscoGAN-M*^{3} arkitekturarenak, eta hirugarrenean, aldiz, *USE* eta *DiscoGAN-M*^{3} sareetako esaldi-bektore bana erabili dira.

5.4.2 Emaitzak

5.10 taulan *BERT* arkitekturarekin egindako esperimentu gainbegiratuen korrelazioak azaltzen dira.

		STS-B*			**vSTS v1.0**	**ALL**
		Train	Dev	Test		
BERT	Classifier	77,0174%	81,2954%	68,9715%	80,4627%	73,6974%
	Regressor	92,7590%	87,9941%	82,9527%	88,9702%	85,9406%

	STS-B			**vSTS v2.0**		
	Train	Dev	Test	Train	Dev	Test
C.	78,9737%	82,3971%	73,5461%	77,1496%	76,3987%	76,5807%
R.	92,8156%	88,9038%	85,2789%	92,0528%	88,1036%	86,9888%

5.10 Taula: *BERT Transformer*ra erabiliz, hiru berfintze pare ezberdin burutu dira. Taulak interpretatzeko adibide bezala, lehenengo taulan *STS-B** multzoaren *Train* eta *Dev* erabili dira ikasketa-prozesuan zehar. Ondoren, sistema ebaluatzeko *STS-B* Test* eta *vSTS v1.0* multzoak erabili dira *Test* bezala.

DiscoGAN-M^3 arkitekturaren lehenengo fasean eraikitzen den G_{I2T} sarearen bidez, irudietatik sortutako goiburukoen errepresentazioak lor daitezke. Ondoren azalduko den bezala, sistemaren ikasketa-prozesua ez da ondo garatu. Hori dela eta, egin diren probetan emaitza txarrak lortu dira. 5.11 taulan bigarren fasea burutu baino lehenagoko emaitzak azaltzen dira. 5.12 eta 5.13 tauletan, aldiz, ikasketa-prozesua bukatu ondorengokoak.

	vSTS v2.0		
	Train	Dev	Test
USE bektoreak	74,7907%	75,8676%	73,6657%
USE & DiscoGAN-M^3 b.	2.2078%	2,5957%	7,3135%
DiscoGAN-M^3 b.	10.9798%	15,7520%	13,7993%

5.11 Taula: Lehenengo lerroan *USE* esaldi-bektoreen arteko kosinu-antzekotasunarekin lortutako korrelazioak azaltzen dira —5.9 taulako parekoa—; bigarrengoan instantzia bakoitzeko *USE* esaldi-bektore bat eta G_{I2T} sareak sortutako beste baten arteko antzekotasunak erabili dira; eta azkenekoan, aldiz, sortzailearen esaldi-bektoreak bakarrik erabili dira.

	vSTS v2.0		
	Train	Dev	Test
USE bektoreak	74,7907%	75,8676%	73,6657%
USE & DiscoGAN-M^3 b.	-2.0991%	-4,8770%	-3,5067%
DiscoGAN-M^3 b.	9,8781%	11,7455%	6,9721%

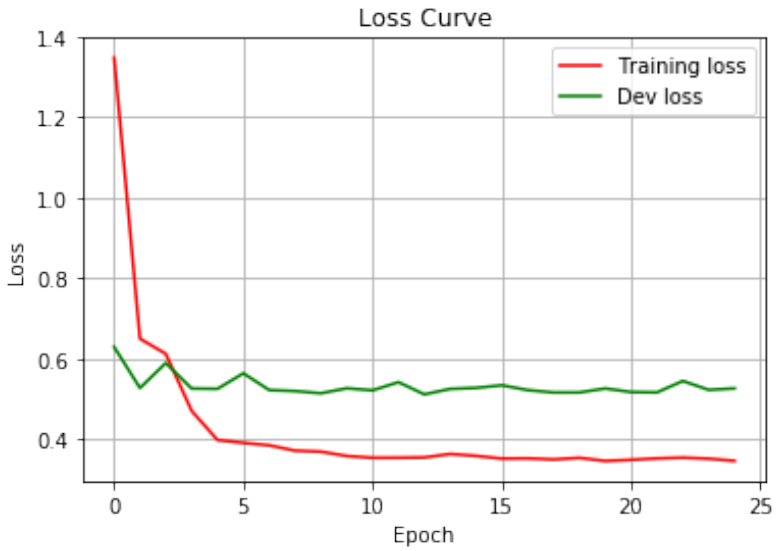
5.12 Taula: Bigarren faseko esaldi pare guztiak erabiliz, esaldi pareen konbinazio ezberdinekin kosinu-antzekotasuna kalkulatu da; eta, ondoren, balio horien eta errealen arteko *Pearson* korrelazioak. Lerro bakoitzak esaldi pareen konbinazio ezberdinak definitzen ditu —5.11 taulan bezala—.

		vSTS v2.0		
		Train	Dev	Test
USE bektoreak		99,3497%	72,8621%	69,6394%
DiscoGAN-M^3 b.		-2,0413%	3,1105%	-1.5442%
Bikote guztiak	*USE* bektoreak	47,7567%	41,3422%	38,2617%
	USE & DiscoGAN-M^3 *b.*	46,2708%	38,0453%	36,6512%
	DiscoGAN-M^3 *b.*	7,4829%	-4,8942%	5,3618%

5.13 Taula: Bigarren fasea burutu ostean lortutako emaitzak azaltzen dira. Lehenengo lerroko aldaeran *USE* esaldi-bektoreak erabili dira bakarrik ikasketa-prozesuan. Bigarrenean G_{I2T} sareak sortu dituenak erabili dira bakarrik. Bukatzeko, azkeneko hiru lerroetan bektore pareen konbinazio guztiak erabili dira entrenatzerako orduan —4. kapituluan proposatutakoa—, baina lerro bakoitzeko emaitzak esaldi-bektore ezberdinen konbinazioak erabiliz lortzen dira —5.12 taulan bezala—.

5.4.3 Ikasketa-prozesuaren detaileak

5.1 irudian *BERT* sarearekin egindako berfintzean burutu den galera-funtzioaren eboluzioa ikus daiteke 25 belaunaldietan zehar. Kasu horretan hamabigarren belaunaldiak dauka *Dev* azpimultzoko galera-balio baxuena. Hori dela eta, belaunaldi horretako pisuak aukeratu dira sistema berfintzerako orduan.

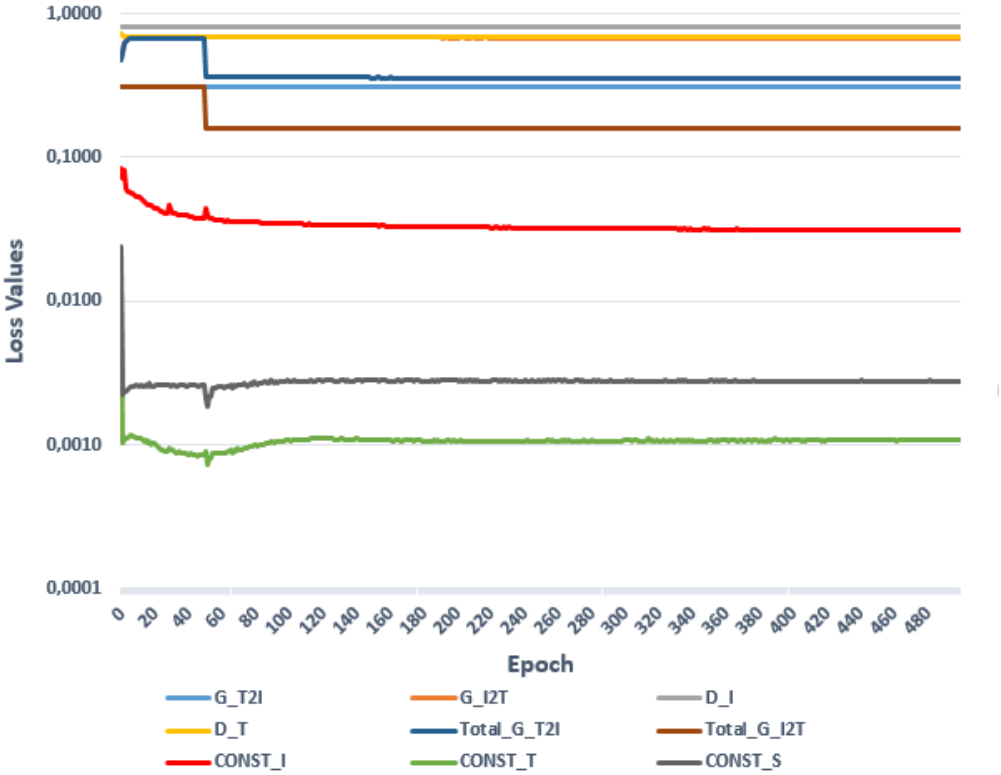


5.1 Irudia: BERT berfintzearen galera-funtzioen eboluzioaren adibidea, 5.10 taula multzoko goiko *STS-B* datu multzoaren taulari dagokiona.

4. kapituluan aipatutako *DiscoGAN-M*^{3} arkitekturaren ikasketa-prozesua *vSTS v2.0* datu multzoa erabiliz burutu da. Ikasketa-prozesuko lehenengo fasean 5.2 irudiko galera-funtzioen balioak lortu dira. Kontuan izan behar da $L_{\text{Total}\cdot\text{G}_{I2T}}$ eta $L_{\text{Total}\cdot\text{G}_{T2I}}$ funtzioen definizioak 60. belaunaldian aldatzen direla, ikasketaren hasieran sortzaileek sortzen dituzten irudiei garrantzi gehiago ematen zaie eta.

Lehen fase honetan zehar ikasketa-tasa ezberdinak probatu badira ere —$[0.0001, 0.1]$ tarteko hainbat ikasketa-tasekin probak eginez—, oro har lortzen diren balioak ia ez dira aldatzen belaunaldietan zehar. Izan ere, berreraikitze-funtzioak —hots, L_{CONST_I}, L_{CONST_T} eta L_{CONST_S} funtzioak— bakarrik aldatzen direla ikus daiteke.

Hasieran ikasketa-indize edo galera-funtzioak gaizki aukeratu edo definitu zirela pentsatu zen; baina, 5.2 irudia ikusiz, *vanishing-gradient* problemaren kasu nabarmen bat dela antzeman da. Izan ere, diskriminatzaileek parte hartzen duten galera-funtzio guztietan ez dira aldaketa nabarmenik azaltzen —diskriminatzaileek parte hartzen duten galera-funtzioak berreraikitze-funtzioak ez direnak dira eta—. Beraz, diskriminatzaileetan zehar



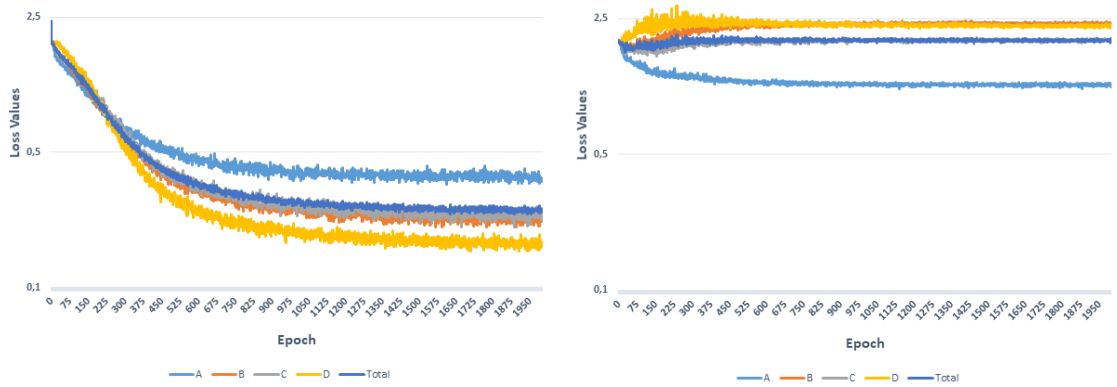
5.2 Irudia: DiscoGAN-M^3-ren lehen faseko galera-funtzioen garapena 500 belaunaldietan zehar. Agertzen diren balioak 4.3, 4.4, 4.5, 4.6 eta 4.9 ekuazioetan definitu diren galera-funtzioekin lortutakoak dira.

atzeranzko propagazioak burutzean aldaketarik sortzen ez direla ondorioztatu daiteke, pisuen gradienteak zerokoak baitira. Kodean gradienteen balioak aztertuz azkeneko hipotesi hau egiaztatu da, baina ez da denborarik izan arazo honen zergatia aurkitu edota konpontzeko.

Lehenengo fasea ondo garatu ez bada ere, bigarren faseko ikasketa burutu nahi izan da —emaitzak zerbait hobetzen diren ikusi ahal izateko—. Guztira, bigarren faseko hiru aldaera probatu dira, ikasketa-tasa eta optimizazio algoritmo berak erabiliz.

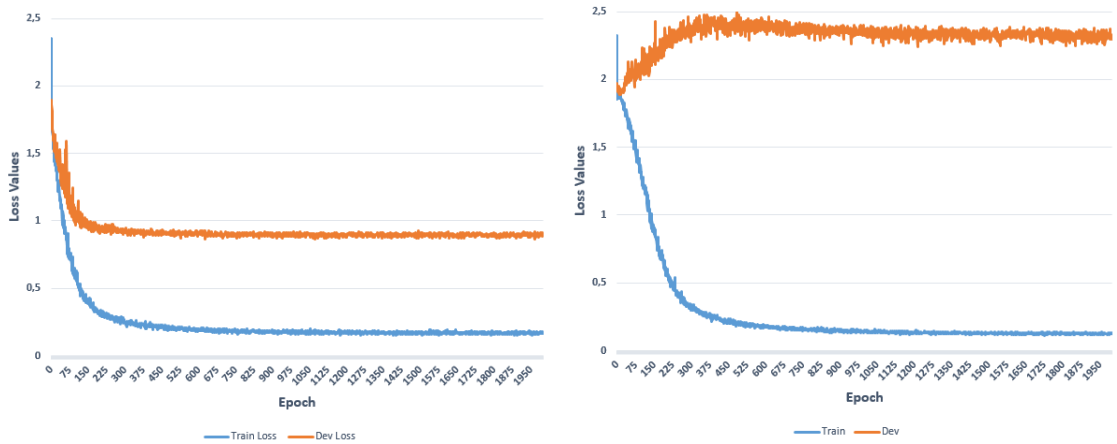
Lehen aipatu den bezala, alde batetik, bigarren fase originala erabili da, 4.6 irudian deskribatu dena eta 4.12 ekuazioko galera-funtzioa erabiltzen duena. Beste aldetik, bi aldaera berri ikasi dira, non esaldi pareen errepresentazioen konbinazio guztiak erabili ordez *USE* sistemarekin lortutakoak edota G_{I2T} sareak sortutakoen arteko antzekotasunak estimatzen diren bakarrik, 4.11 ekuazioetako L_A eta L_D galera-funtzioak erabiliz, hurrenez hurren.

5.3 eta 5.4 irudietan bigarren faseko hainbat aldaeretan lortu diren galera-balioak aztertzen dira.



(a) **(b)**

5.3 Irudia: Bigarren fase originaleko galera-funtzioen garapena. Ezkerreko irudian *Train* azpimultzoaren L_A, L_B, L_C, L_D eta L_{Total} balioak azaltzen dira —4.11 eta 4.12 ekuazioetan definituta—. Eskuinekoan, aldiz, *Dev* azpimultzoko balio berak azaltzen dira. Bi grafikoak eskala logaritmiko berdina dute ardatz bertikalean.



(a) **(b)**

5.4 Irudia: Bigarren faseko aldaeren galera-funtzioen garapena. Ezkerreko irudian *Train* eta *Dev* azpimultzoen L_A balioak azaltzen dira —4.11 ekuazioan definituta—, esaldi pare originalekin kalkulatzen dena. Eskuinekoan, aldiz, azpimultzo bereko L_D balioen eboluzioa azaltzen da, sortutako esaldi pareekin kalkulatzen dena. Bi grafikoak eskala bera erabiltzen dute ardatz bertikalean.

5.3 eta 5.4 irudietako galera-funtzioak aztertuz, G_{I2T} sareak sortu dituen esaldi-bektoreen arteko antzekotasunak orokortzeko zailtasunak dituela ikus daiteke. Izan ere, *Train* azpimultzoan galera-balioak txikitzen badoaz ere, *Dev* azpimultzoak ez du tendentzia bera jarraitzen.

Bide batez, ikasketa-prozesuaren exekuzio-denborak 5.14 taulan azaltzen dira.

		Denborak
1. Fasea		5 d 5 h 1 min 42.103 s
2. Fasea	Pare guztiak	4 h 39 min 55.509 s
	USE bektoreak	2 h 11 min 56.65 s
	DiscoGAN-M^3 b.	4 h 33 min 33.572 s

5.14 Taula: Ikasketa-prozesuaren fase ezberdinen exekuzio-denborak.

5.5 Esperimentuen hausnarketa

Azter ditzagun orain datu multzo eta sistema ezberdinen errendimenduak proba ez-gainbegiratuetan. 5.3, 5.4 eta 5.5 tauletan, *BERT*, *GloVe* eta *GPT-2* sistemak erabili dira datu multzo ezberdinen gainean. Alde batetik, honek sistema bakoitzaren errendimendua aztertzeko balio du; eta bestetik, datu multzoen zailtasuna intuitiboki antzematea ahalbidetzen du. Izan ere, sistema guztiek datu multzo batean emaitza okerragoak lortzen baditu, datu multzo horren instantziak, oro har, zailagoak direla esan nahi du.

Aurreko tauletara bueltatuz, proba ez-gainbegiratuetan erabilitako modalitate bakarreko sistema guztietan, *vSTS v1.0* datu multzoan [0.04, 0.21] tarteko korrelazio hobekuntzak ikusten dira beste bi datu multzoekin konparatuz. *vSTS v2.0* eta *STS-B* datu multzoen artean, aldiz, ez dago ezberdintasun nabarmenik, emaitzak batean bestean baino hobeak izanik batzuetan.

Alde batetik, honek 3. kapituluan aipatu den *vSTS v1.0* datu multzoaren instantzien erraztasuna baieztatzen du, baita 4. kapituluan datu multzoko *v2.0* bertsioaren instantzia zailagoak ondo hautatu direla ere. Beste aldetik, *STS-B* datu multzoan aurki daitezkeen esaldietan *vSTS v2.0* multzoan baino barietate handiagoa dagoela esan bada ere, honek ez du emaitzetan diferentzia handirik suposatu egin diren esperimentu ezberdinetan.

Erabilitako sistemen aldetik, zentroideak kalkulatzerako orduan hitz hutsak baztertzean lortutako korrelazioak handiagoak direla ikus daiteke —hitz hutsek zarata bakarrik gehitzen dute eta—. Sistema ezberdinen errendimenduan, behintzat, *BERT*-ek ematen ditu

hiruetatik emaitza hoberenak, *GloVe* gertu dabilelarik. *GPT-2* sareak esperotakoak baino emaitza kaxkarragoak ematen ditu, testu-sorkuntzarako diseinatuta dagoelako ziurrenik. Azkenik, aipatzekoa da *BERT* sisteman *CLS* geruzako errepresentazioak ez dutela esaldien esanahi semantiko nabarmenik gordetzen, hauen korrelazioak 0 ingurukoak baitira.

Aurreko emaitzak *VSE++* sistemaren aurrean oso baxuak dira —5.6, 5.7 eta 5.8 taulak—. Izan ere, irudien errepresentazioen arteko kosinuaren antzekotasunekin lortutako korrelazioak ez dira esaldienak bezain onak; eta, hala ere, irudien adierazpenekin lortutako emaitzak modalitate bakarreko sistemetan lortu diren emaitza hoberenen parean daude.

Esaldien errepresentazioak erabiltzen badira, aldiz, aurreko korrelazioak [0.15, 0.2] puntutan igotzen dira, proba ez-gainbegiratuetan emaitza hoberenak lortuz. Esaldiak eta irudiekin lortutako antzekotasunen batez-bestekoa erabiltzen bada, emaitzak esaldien antzekotasunak bezain onak dira. Azkenik, *ResNet152* konboluzio-sarea erabiltzean, esaldien arteko antzekotasunen korrelazioak *VGG19* erabiliz baino baxuagoak direla; baina, irudien arteko antzekotasunen korrelazioak zerbait handiagoak bihurtzen dira ikus daiteke —ezberdintasunak ia nabaritzen ez badira ere—.

*vSTS v2.0** datu multzoa *VSE++* sisteman erabiltzeko sortu da, *VSE++* sistemaren ikasketa-prozesuan *MS-Coco* erabili baita —*MS-Coco* multzoko instantziak erabiltzeak korrelazioetan eragina duen ikusi nahi da eta—. *MS-Coco* gabeko *vSTS v2.0* bertsio honek *vSTS v2.0* datu multzoak bezain korrelazio onak —eta hobeagoak— dituenez, *VSE++* esaldipareen antzekotasuna aztertzeko sistema sendo bat dela esan daiteke.

USE sistema erabiltzean berfintzerik burutu ez bada ere, burututako proba ez da guztiz ez-gainbegiratua, *STS* ataza ikasketa-prozesuan ikusi baitu. Sistemak lortzen dituen korrelazioak modalitate bakarreko sistemek proba ez-gainbegiratuetan lortutakoak baino zerbait hobeagoak dira —5.9 taula—. Lortutako korrelazio altuak direla eta, sistema honek sortzen dituen esaldi-bektoreak erabili dira *DiscoGAN-M*^{3} arkitekturan.

Azkenik, *DiscoGAN-M*^{3} sistemaren errendimendua aztertu baino lehen, *BERT* sistemaren berfintzea laburki komentatzea falta da. Arkitektura honek proiektuko emaitza onenak lortu ditu erregresio-geruza erabili deneko hiru kasuetan —5.10 taula—. Sailkatzailearen aldaerak ez du hain errendimendu ona izan; baina, hala ere, sailkatzailearen emaitzak *USE* sistemarekin lortu direnak baino zerbait hobeagoak izan dira. *STS-B** datu multzoa ikasketa-prozesuan zehar erabili da *vSTS v1.0 Test* azpimultzo bezala erabili ahal izateko —*STS-B* datu multzoan *vSTS v1.0* multzoko instantziak daude eta—, eta, kasu honetan, emaitzak espero bezain onak izan dira ere bai.

DiscoGAN-M^3 arkitekturarekin zerikusia duten esperimentuek ez dituzte emaitza onik eman. Hau ikasketa-prozesu oker baten ondorioa da; eta, beraz, sistema honek duen ahalmena ezin izan da guztiz neurtu. 5. kapituluan, *vanishing-gradient* arazoaz hitz egin da, baina hau ez da arkitekturaren arazo bakarra.

Sortzaile eta diskriminatzaileen konplexutasun txikiak sistemak egin dezakeena mugatzen du. Arkitektura honen ahalmena neurtu ezin izan bada ere, modalitate anitzen arteko mapaketak *AlexNet* sareak eman dezakeena baino konplexutasun handiagoa behar duela susmatzen da.

Gainera, modalitate ezberdineko errepresentazioek galera-funtzioan duten eragin ezberdina kontuan hartzea ideia ona izango litzateke. 5.2 irudian berreraikitze-funtzioetan arreta jartzen bada, irudien arteko berreraikitze-funtzioen balioak testuena baino 10-50 aldiz handiagoa dela ikus daiteke. Beraz, atzeranzko propagazioa aplikatzean irudien berreraikitzeari lehentasun handiagoa ematen zaio —testuarenarekin konparatuz behintzat—.

DiscoGAN-M^3 sistemaren ikasketa-prozesua ez denez zuzena izan, ez du merezi lortu diren korrelazioak gehiago komentatzea, baina irudiak zerbait berreraikitzen ikasi duenez, 5.5 irudi multzoan lehen fasean zehar lortu diren hobekuntzak antzeman daitezke —sortutako azkeneko irudiak oso onak ez badira ere, hobekuntza bat ikus daiteke—.

5.5 irudi multzoan sistemak duen beste arazo bat antzeman daiteke. Izan ere, $\mathbf{x}_I$ sarrerako iruditik $\mathbf{x}_{ITI}$ lortzean antzeko irudi bat izatea espero daiteke —4. kapituluan esan den bezala, funtzio bijektibo bat eta bere alderantzizkoa aplikatzean $\mathbf{x}_{ITI}$ lortu beharko litzateke—. Hala ere, $\mathbf{x}_{IT}$ irudia lortzeko G_{I2T} sareak ausazko goiburuko baten esaldi-bektorea izan beharko luke sarrera-bektore bezala —esanahi semantiko oso ezberdina izango lukeena—. Antzeman daitekeenez, $\mathbf{x}_{IT}$ irudiak $\mathbf{x}_{ITI}$-ren oso parekoak dira. Beraz, G_{T2I} sortzaileak irudia berreraikitzeko gehienbat $\mathbf{f}_T$ bektorea erabiltzen duela esan nahi du honek, $\mathbf{t}_T$ esaldi-bektorea ia kontuan hartu gabe.

Arkitektura berri honekin emaitza dezenterik lortu ez bada ere, egin diren esperimentuek sistemak dituen hainbat arazo erakutsi dizkigu; etorkizunean arkitektura konpontzeko argibide bezala erabili daitezkeelarik.

5.5 Irudia: DiscoGAN-M^3 sareak lehenengo ikasketa fasean zehar sortutako irudiak hainbat belaunaldietan zehar. Lerro bakoitzean ikasketa-prozesuko belaunaldia azaltzen da. Irudi bakoitzeko, $\mathbf{x}_{TI} = G_{T2I}(\mathbf{x}_T)$ eta $\mathbf{x}_{ITI} = G_{T2I}(G_{I2T}(\mathbf{x}_I))$ azaltzen dira, non $\mathbf{x}_T = \mathbf{f}_T \oplus \mathbf{t}_T$ bektorea $\mathbf{x}_I$ irudiaren $\mathbf{f}_{IT}$ bektoreaz osatuta dagoen —kasu honetan $\mathbf{f}_T = \mathbf{f}_{IT}$ izanik—, baita irudiari ez dagokion goiburuko baten $\mathbf{t}_T$ bektoreaz ere.

6. KAPITULUA

Ondorioak

Proiektu honetan modalitate anitzeko sistemen estimazioak modalitate bakarra erabiltzen dutenekin konparatzea izan da helburu nagusia. Hainbat esperimentu burutu dira *vSTS* eta *STS-B* datu multzoen gainean erabili diren sistemek dituzten errendimenduak aztertzeko. Gainera, sare sortzaile antagonikoetan oinarritutako arkitektura berri bat diseinatu da, testu eta irudien errepresentazioen arteko mapaketa burutzeko gai izan beharko lukeena. Hala ere, azken hau ezin izan da aztertu egindako esperimentuetan, sistemaren ikasketa-prozesua ez baita ondo garatu.

Oro har, egin diren probak bi ataletan banatu daitezke: gainbegiratuetan eta ez-gainbegiratuetan, hain zuzen ere.

Alde batetik, emaitza gainbegiratuetan modalitate anitzak erabiliz ez dira emaitza onak lortu, erabili den sistema bakarrak ataza ez duelako ondo ikasi. Gainera, modalitate bakarreko *BERT* sistemaren berfintzean lortutako emaitzak proiektuko hoberenak izan dira, [0.85, 0.87] arteko korrelazioak lortu baitira —5.10 taulan lortzen diren emaitzak artearen egoera definitzen dutelarik *vSTS* datu multzoko bi bertsioetan—.

Hala ere, beste aldetik, modalitate bakarreko proba ez-gainbegiratuetan 0.7 baino korrelazio handiagoak ez dira lortu —*vSTS v1.0* datu multzoan izan ezik, instantzia errazez osatuta dago eta—; baina, modalitate anitzeko *VSE++* sistemaren kasuan [0.81, 0.83] inguruko korrelazioak lortu dira —5.6, 5.7 eta 5.8 taulak—, sistema gainbegiratu hoberenetik oso gertu eta modalitate bakarreko sistemen proba ez-gainbegiratuetan baino emaitza askoz hobeagoak lortuz.

Kasu hauetan, *image-caption retrieval* bezalako modalitate anitzeko ataza ikasketa-trans-

ferentzia burutzeko erabiltzeak duen abantaila erakusten du, emaitza oso sendoak itzuliz —hizkuntza-ereduetan oinarritutako sistemetan ez bezala—.

6.1 Etorkizunerako lanak

Esperimentu dezente egin badira ere, hainbat esperimentu bertan behera utzi behar izan dira denbora-falta dela eta. Alde batetik, *VSE++* sistemaren gainean berfintze bat burutzea ondo etorriko litzake. Izan ere, esperimentu ez-gainbegiratuetan lortu dituen emaitzak oso onak dira, ia *BERT* sistemak proba gainbegiratuetan definitu duen artearen egoeraren parekoak. Horregatik, berfintze horretan lortzen diren korrelazioak *BERT* sistemarenak bezain onak, hobeak ala okerragoak diren aztertzea ondo egongo litzateke. Hasiera batean hau burutu nahi bazen ere, arkitektura berria garatzeari garrantzi handiagoa eman zitzaion, eta azkenean proba hau bertan behera utzi zen.

Gaur egun hizkuntzaren-prozesamenduko arloa pil-pilean dago, edozein momentutan aurrekoak baino arkitektura berri hobeagoak agertzen baitira. 2. kapituluan, *XLNet* arkitektura aipatu da [22]. *Google*-n *Transformer* berri hau hainbat atazetan *BERT* baino hobea dela aitortu dute, *STS* atazan barne. Hori dela eta, arkitektura berri honekin lortzen diren korrelazioak *BERT* sistemarekin —eta beste arkitekturekin— lortu direnak konparatu beharko lirateke.

Azkenik, sortu den *DiscoGAN-M*^{3} arkitektura konpondu beharko litzateke. Ideiak potentziala badu ere, bere garapena ez da esperotakoa bezain ona izan. Hainbat arazo aurkitu dira probak egin diren heinean, eta batzuk konpondu badira ere, beste hainbat hor gelditu dira konpontzeko zain. Sarearen ikasketa-prozesuaren *vanishing gradient* problema, sareen konplexutasuna eta galera-funtzio ezberdinak aztertzea etorkizunerako gelditzen dira.

Galera-funtzioen inguruan zerbait aipatzearren, adibidez, ikasketa-prozesuko bigarren fasean batez-besteko errore koadratikoa —4.11 ekuazioa— aldatu daiteke, *Pearson*en korrelazioan oinarritutako galera-funtzio batekin ordezkatuz. Sareak ausaz aukeratutako *mini-batch* batean estimatutako antzekotasunak $\mathbf{x}$ bektorean badaude, eta horien balio errealak $\mathbf{y}$ bektorean, 6.1 ekuazioan galera-funtzio berri hori azaltzen da, n elementu kopurua izanik.

$$L_A = 1 - \frac{1}{n-1} \sum_{i=1}^{n} \left(\frac{\mathbf{x}_i - \overline{\mathbf{x}}}{\sigma_{\mathbf{x}}} \right) \left(\frac{\mathbf{y}_i - \overline{\mathbf{y}}}{\sigma_{\mathbf{y}}} \right) \tag{6.1}$$

Galera-funtzio berri honetan sarea ez da antzekotasun errealak estimatzen saiatzen, *mini-batch* horien antzekotasunak ordenatzen baizik. Honetarako, gero eta *mini-batch* handiagoak erabiltzeak, orduan eta emaitza zuzenagoak kalkulatzea ekarriko du, instantzia gehiago kontuan hartuko direlako galera-funtzioaren aplikazio bakoitzean. Horrela, lortzen den korrelazioa oso altua bada, emaitzak $[0,5]$ tartera pasa ahalko lirateke $y = m \cdot x + n$ motako ekuazio lineal bat ebatziz. Aldaketa asko burutu daitezke, eta baliteke honek hobekuntza handirik ez ematea, baina sarearen emaitzei askatasuna emateko egokia izan daiteke —balio definitu batzuk itzultzea ez duelako derrigortzen, ordena bat jarraitzea baizik—.

Aipatu diren hobekuntza hauek doktorego-tesian lantzea espero da, gaian gehiago sakonduz eta modalitate anitzeko beste ataza batzuk aztertuz —galdera-erantzute bisuala edo *Visual Question-Answering*, besteak beste—.

Bibliografia

[1] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.

[2] M. Pagliardini, P. Gupta, and M. Jaggi, "Unsupervised learning of sentence embeddings using compositional n-gram features," *CoRR*, vol. abs/1703.02507, 2017.

[3] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia, "SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, (Vancouver, Canada), pp. 1–14, Association for Computational Linguistics, Aug. 2017.

[4] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, (Lisbon, Portugal), pp. 632–642, Association for Computational Linguistics, Sept. 2015.

[5] O. Lopez de Lacalle, A. Soroa, and E. Agirre, "Evaluating multimodal representations on sentence similarity: vsts, visual semantic textual similarity dataset," *CoRR*, vol. abs/1809.03695, 2018.

[6] C. Olah, "Understanding lstm networks." https://colah.github.io/posts/2015-08-Understanding-LSTMs/, 2015. Accessed: 2015-08-27.

[7] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[8] F. Gers and J. Schmidhuber, "Recurrent nets that time and count," in *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN'00)-Volume 3 - Volume 3*, vol. 3, pp. 189 – 194 vol.3, 02 2000.

[9] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," in *International Conference on Machine Learning*, pp. 2342–2350, 2015.

[10] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "Lstm: A search space odyssey," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 10, pp. 2222–2232, 2016.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.Ñ. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017.

[12] J. Alammar, "The illustrated transformer." https://jalammar.github.io/illustrated-transformer/, 2018. Accessed: 2018-06-27.

[13] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[14] D. Cer, Y. Yang, S. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, Y. Sung, B. Strope, and R. Kurzweil, "Universal sentence encoder," *CoRR*, vol. abs/1803.11175, 2018.

[15] M. Iyyer, V. Manjunatha, J. Boyd-Graber, and H. Daumé III, "Deep unordered composition rivals syntactic methods for text classification," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, vol. 1, pp. 1681–1691, 2015.

[16] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *CoRR*, vol. abs/1310.4546, 2013.

[17] D. C. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," *CoRR*, vol. abs/1202.2745, 2012.

[18] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018.

[19] A. Radford, K.Ñarasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*, 2018.

[20] M. E. Peters, M.Ñeumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *CoRR*, vol. abs/1802.05365, 2018.

[21] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M.Ñorouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, "Google's neural machine translation system: Bridging the gap between human and machine translation," *CoRR*, vol. abs/1609.08144, 2016.

[22] Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *CoRR*, vol. abs/1906.08237, 2019.

[23] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, 2019.

[24] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," *CoRR*, vol. abs/1508.07909, 2015.

[25] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "GLUE: A multi-task benchmark and analysis platform for natural language understanding," *CoRR*, vol. abs/1804.07461, 2018.

[26] A. Wang, Y. Pruksachatkun, N.Ñangia, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "Superglue: A stickier benchmark for general-purpose language understanding systems," *CoRR*, vol. abs/1905.00537, 2019.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), pp. 1097–1105, Curran Associates, Inc., 2012.

[28] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, June 2009.

[29] Y. Zhang and B. C. Wallace, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," *CoRR*, vol. abs/1510.03820, 2015.

[30] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv 1409.1556*, 09 2014.

[31] S. C. Arishanapally, "Building vgg19 with keras." https://medium.com/@saicharanars/building-vgg19-with-keras-f516101c24cf, 2019. Accessed: 2019-04-16.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.

[33] F. Faghri, D. J. Fleet, J. R. Kiros, and S. Fidler, "VSE++: improved visual-semantic embeddings," *CoRR*, vol. abs/1707.05612, 2017.

[34] S. Seog Han, G. Hun Park, W. Lim, M. Shin Kim, J.-I. Na, I. Park, and S. Eun Chang, "Deep neural networks show an equivalent and often superior performance to dermatologists in onychomycosis diagnosis: Automatic construction of onychomycosis datasets by region-based convolutional deep neural network," *PLOS ONE*, vol. 13, p. e0191493, 01 2018.

[35] R. Kiros, R. Salakhutdinov, and R. S. Zemel, "Unifying visual-semantic embeddings with multimodal neural language models," *CoRR*, vol. abs/1411.2539, 2014.

[36] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014.

[37] X. Chen, H. Fang, T. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick, "Microsoft COCO captions: Data collection and evaluation server," *CoRR*, vol. abs/1504.00325, 2015.

[38] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descrip-

tions," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 67–78, 2014.

[39] M. Hodosh, P. Young, and J. Hockenmaier, "Framing image description as a ranking task: Data, models and evaluation metrics," *J. Artif. Int. Res.*, vol. 47, pp. 853–899, May 2013.

[40] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27* (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds.), pp. 2672–2680, Curran Associates, Inc., 2014.

[41] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," *CoRR*, vol. abs/1703.05192, 2017.

[42] J. Allen, *Natural language understanding*. Pearson, 1995.

[43] A. P. Parikh, O. Täckström, D. Das, and J. Uszkoreit, "A decomposable attention model for natural language inference," *CoRR*, vol. abs/1606.01933, 2016.

[44] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 12 2014.

[45] N. S. Keskar and R. Socher, "Improving generalization performance by switching from adam to SGD," *CoRR*, vol. abs/1712.07628, 2017.

[46] V. Zhelezniak, A. Savkov, A. Shen, and N. Y. Hammerla, "Correlation coefficients and semantic textual similarity," *CoRR*, vol. abs/1905.07790, 2019.

Eranskinak

Erabilitako terminologia eta laburdurak

- **Aktibazio-funtzioa**: Activation function
- **Agerkidetza**: Co-ocurrence
- **Atentzio-mekanismoa**: Attention-mechanism
- **Atzeranzko propagazioa**: Back-propagation
- **Aurrerantz-elikatutako neurona-sarea**: Feedforward neural-network
- **Aurreranzko propagazioa**: Forward-propagation
- **Bana-banako korrespondentzia**: One-to-one correspondence
- **Batez-besteko errore koadratikoa**: Mean Square Error, MSE
- **Berfinketa**: Fine-Tuning, FT
- **Bideo- eta testu-lerrokatzea**: Video and text alignment
- **Biderketa eskalarra**: Dot product
- **Bi noranzkoko transformerra**: Bidirectional transformer
- **Deskodetzailea**: Decoder
- **Egoera-gelaxka**: Cell-state
- **Esaldi-bektorea**: Sentence-embedding
- **Funtzio logistikoa**: Logistic function
- **Galdera-erantzute bisuala**: Visual Question-Answering, VQA

- **Galera-funtzioa**: Loss-function
- **Gelaxka**: Cell
- **Geruza-dentsoa**: Dense-layer
- **Goiburukoa**: Caption
- **Helburu-funtzioa**: Objective-function
- **Hitz-bektorea**: Word embdding
- **Hitz hutsa**: Stop Word, SW
- **Hitz-zaku**: Bag-of-Words, BoW
- **Hizkuntza-eredua**: Language Model, LM
- **Hizkuntzaren Prozesamendua, HP**: Natural Language Processing, NLP
- **Hizkuntzaren Ulermena, HU**: Natural Language Understanding, NLU
- **Ikasketa-tasa**: Learning-rate
- **Irudi edo goiburuko berreskurapena**: Image-caption retrieval
- **Irudien goiburuko sorkuntza**: Caption generation
- **Itzulpen automatikoa**: Machine-Translation, MT
- **Katearen erregela**: Chain-rule
- **Kodetzailea**: Encoder
- **Konboluzio iraulia**: Transposed-convolution
- **Konboluzio-sarea**: Convolutional Neuronal-Network, CNN
- **Konputagailu bidezko ikusmena**: Computer Vision, CV
- **Kosinu-antzekotasuna**: Cosine-similarity
- **Modalitate anitza**: Multimodal
- **Neurona-sarea**: Neural-Network, NN

- **Neurona-sare errepikakorra**: Recurrent Neural-Network, RNN
- **Objektuen antzematea**: Object-recognition
- **Posizio-kodeketa**: Positional-encoding
- **RNN gelaxka hedatu**: Unfold *RNN* cell
- **Sare sortzaile antagonikoa**: Generative Adversarial Network, GAN
- **Testu eta irudi bidezko antzekotasun semantikoa**: Visual Semantic Textual Similarity, vSTS
- **Testuen antzekotasun semantikoa**: Semantic Textual Similarity, STS
- **Transferentzia ikasketa**: Transfer-learning
- **Zarata-bektorea**: Noise-vector