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

Informatika Ingeniaritzako Gradua

Konputazioa

Gradu Amaierako Lana

Ezagutza baseak itzultzaile neuronaletan

Egilea

Jesús Javier Calleja Pérez

2020

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

Informatika Ingeniaritzako Gradua

Konputazioa

Gradu Amaierako Lana

Ezagutza baseak itzultzaile neuronaletan

Egilea

Jesús Javier Calleja Pérez

Zuzendariak

Gorka Labaka Intxauspe

Olatz Pérez de Viñaspre Garralda

Ander Soraluze Irureta

Laburpena

Proiektu honetan itzulpen automatikoko sistemetan kanpoko ezagutza sartzea aztertzen da gaur egungo arloaren egoeraren arkitekturan, Transformerran, itzulpenak hobetzeko asmoz. Testuetan maiz agertzen ez diren hitzak, domeinu jakin bateko hitz eta termino teknikoak izan ere, itzultzea zaila da itzultzaile automatikoentzat. Horregatik, kanpoko ezagutza sartzea onuragarria da emaitza hobeak lortzeko. RNNetan erabili diren hurbilpenak aztertzen dira eta Transformerrarentzako berri bat diseinatzen da, arkitektura honek erabiltzen duen teknologiaz aprobetxatuz, autoarreta. Emaitzak ebaluatzeko domeinu biomedikoko testuen itzulpenen kalitatea aztertzen da. SNOMED CT ontologia klinikokoak batzen dituen terminoak eta hemendik ikasitako termino hauen errepresentazioak, *word embeddingak*, erabiltzen dira kanpoko ezagutza bezala.

Gaien aurkibidea

Irudien aurkibidea

Taulen aurkibidea

1. KAPITULUA

Sarrera

Hizkuntzaren Prozesamenduaren arloaren barruan, lan hau itzulpen automatikoan kokatzen da. Itzulpen automatikoa software bitarteko hizkuntza batetik bestera testu bat itzultzeko prozesua da. Gero eta globalizatuago dagoen munduan, ezinbestekoa da teknologia hau profitatzen duten programak erabiltzea hizkuntza desberdinetan dauden testuak ezagutzen den batera arin itzultzeko.

Kasu honetan, testu klinikoko itzultzaile automatiko bat aztertzen eta hobetzen da arlo horretan egon daitezkeen beharrak asetzeko. Adibidez, Euskal Autonomia Erkidego mailan, sistema zentralizatuta dagoenez eta edozein profesional gaixo baten historial klinikora sar daitekeenez, denek uler dezaten gaztelaniaz idazten dira. Praktikan, profesionalek esfortzu estra egin behar dute gaztelaniaz egiten ez diren kontsultekin, Euskal Herrian esate baterako, euskaraz egiten ari den kontsultak gaztelaniara itzuliz eta gaixoaren jarraipen egokia bermatuz.

Garai honetan itzultzaile automatikoen garapena handitu bada ere, oraindik itzultzaileentzako zaila da osasunaren arloan dauden beharrak asetzea, zehazki, profesionalek erabiltzen duten lengoai laburra eta teknikoa itzultzeko ezgaitasuna. Testuinguru klinikoan erabiltzen den terminologia hain zehatza eta zabala da, ezen ezta itzultzaile automatiko berrienak ere zehazki ikasteko gai diren. Horregatik, beharrezkoa da kanpoko ezagutza baseen eransketa egitea, hiztegi elebidun espezializatuak edo ontologia eleanitzak bezala, besteak beste.

Lan honetan, beste arkitekturetan kanpoko ezagutza sartzeko teknikak aztertzen dira, gaur egun itzultzaile automatikoen artearen egoeran dagoen arkitekturan egokitzeko eta au-

tomatikoki ebaluatuko da corpus kliniko baten gainean. Horretaz gain, kanpo ezagutza bezala aukeratuko da ezagutza base bat arkitekturan integratzeko.

Txosten hau sarrera honekin hasten da, lanaren motibazioa deskribatuz eta egingo dena labur azalduta. Ondoren Proiektuaren Helburuen Dokumentua deritzon atala dago. Bertan lanaren irismena agertzen da eta hau burutzeko egin den plangintza deskribatzen da. Aurrekarien atalean lan hau ondo ulertzeko beharrezkoak diren oinarrizko azalpenak ematen dira itzultzaile automatikoek urteetan zehar izandako garapenari buruz, gaur egun itzulpen automatikoko atazarako erabilitako sare neuronaleei buruz, hitz errepresentazioari buruz eta ezagutza baseei buruz. Metodologian lanean zehar burututako atazak azaltzen dira: corpusen aukeraketa, hauen aurreprozesaketa, SNOMED CT ontologiaren terminoen embeddingen entrenamendua, ereduaren diseinua itzulpen automatikoa gauzatzeko eta inplementazioaren xehetasunak. Amaitzeko, ereduak emandako emaitzak ematen dira eta hauetatik lortutako ondorioak azaltzen dira, etorkizuneko lanarekin eta plangintzaren jarraipen eta kontrolarekin batera.

2. KAPITULUA

Proiektuaren Helburuen Dokumentua

Atal honetan, proiektuaren bideragarritasuna justifikatzen duten elementuak agertzen dira; hala nola, plangintza, lan metodologia eta arriskuen kudeaketa.

2.1 Proiektuaren deskribapena eta irismena

Proiektu honetan itzultzaile automatiko bat hasieratik inplementatzen da, [Vaswani et al., 2017]-ek proposatutako Transformer arkitektura jarraituz, eta kanpoko informazioa sartzeko moldaketa egiten zaio emaitzak hobetzeko asmoz. Sistemak SNOMED CT ontologia klinikotik hartuko du kanpoko informazioa aurreko inplementazioan egindako moldaketaren bitartez. Horretaz gain, kanpoko ezagutza ereduak erabiltzeko beharrezkoa den prozesua egiten da. Amaitzeko, sistema ebaluatzeko domeinu biomedikoko testuak erabiliko dira.

2.2 Proiektuaren plangintza

Atal honetan, lanaren atazen identifikazioa eta plangintza egiten da.

2.1 Irudia: LDE diagrama.

2.2.1 Lanaren banaketa

Lana garatzeko beharrezkoak diren atazak identifikatzeko, lanaren deskonposaketa egitura (LDE) egiten da 2.1 irudian.

LDE diagraman agertzen diren lan paketeen deskribapena jarraian azaltzen dira eta bakoitzaren aurreikusitako denbora 2.2 taulan agertzen da.

- **Arloaren egoera.** Ataza honetan proiektuarekin erlazionatuta dagoen arloaren egoera aztertzen da.
  - **Itzultzaile automatikoen arloaren egoera.** Urteetan zehar, itzultzaile automatikoak garatzeko erabili diren teknologiak aldatuz joan dira, emaitzak gero eta hobeak lortuz. Horregatik, beharrezkoa da itzultzaile automatikoen gaur egungo egoera ikertzea, baita aurreko teknologiak ere, arloaren ideia nagusia izateko eta teknologien garapenaren zergatia ulertzeko.
  - **Kanpoko ezagutza sartzeko teknikak.** Ezagutza baseak ereduan integratze aldera, beste arkitekturekin erabili diren teknikak aztertzen dira eta horien ideietan oinarrituz esperimentuaren diseinua egin daiteke.

- **Esperimentazioa.** Ataza honetan, behin arloaren egoera aztertuta, ereduaren arkitekturaren aukeraketa, honetan egingo diren hobekuntzen diseinua, ezagutza basearen eta corpusen aukeraketa, inplementazioa, lortu diren emaitzak eta hauen baloraziOa egiten da.

  - **Diseinua.** Arloaren egoerako azterketa abiapuntutzat izanik, esperimenturako erabiliko den arkitektura aukeratzen da, berorrek dauzkan ezaugarriak kontuan hartuz hobekuntza planteatzen da eta erabiliko den ezagutza basea aukeratzen da hautatutako corpusaren domeinua aintzat hartuta.
  - **Inplementazioa.** Itzultzaile automatikoaren ereduaren diseinuaren inplementazioa gauzatzen da ataza honetan, aukeratutako lengoaia eta honek lanerako erabilgarriak diren liburutegiak erabiliz. Ataza honetan corpusaren eta ezagutza basetik datozen datuen garbiketa edota moldaketa ere burutzen da.
  - **Emaitzak.** Corpusean egindako proba desberdinen emaitzak batzen dira ataza honetan. Proba hauek baliagarriak izango dira hobekuntzaren kalitatea neurtzeko eta ebaluatzeko, corpusa itzultzerako orduan baldintza desberdinetan jarriz eta modeloaren ahalmena neurtuz.
  - **Ebaluazioa.** Emaitzen ebaluazioa beharrezkoa da lanaren ondorioak lortzeko eta hobekuntza onuragarria izan denetz baloratzeko.

- **Emangarriak.** Ataza honetan, lanaren emangarriekin lotuta dauden eginkizunak batzen dira: memoriaren idazketa, lanaren deskribapena eta garapena osoa batzen duen dokumentu hau bera, eta aurkezpena, defentsa egunean memoria honen atal garrantzitsuenak biltzen dituzten azalpenak eta gardenkiak.

  - **Memoria.** Ataza honetan memoriaren edukiak eta egitura planteatzen da eta honen idazketa burutzen da. Memoriaren eduki nagusiak lanaren plangintza, burututako ikerketa eta diseinatutako eta egindako esperimentuen nondik norakoak azaltzen dira, hauen emaitzekin, ondorioekin eta etorkizuneko lanarekin batera.
  - **Aurkezpena.** Behin memoria idatzita, tribunalaren aurrean egindako lanaren berri emateko aurkezpena presatzen da: emango diren azalpenak eta hauek euskarri bezala izango dituzten gardenkiak.

- **Kudeaketa.** Ataza honetan, lana egitea bermatzen duten eginkizunak batzen dira: plangintza, lanaren helburuen bideragarritasuna justifikatzen duena, eta jarraipen

eta kontrola, atazak plangintzaren arabera edo plangintzak posible ikusten dituen desbiderapenen barruan egiten ari direla bermatzen duena.

- **Plangintza.** Atal honetan, lanaren irismena eta hauek betetzeko egin behar diren atazak zehazten dira, existitzen diren mugarriak, denbora-murriztapenak eta arriskuak kontuan hartuz.
- **Jarraipena eta kontrola.** Plangintza errespetatzen dela kontrolatzen da atal honetan; arriskuren bat egonez gero, arintze neurriak hartzeko ere balio duena.

2.2.2 Mugarriak

Lan honek dituen emangarrien mugarriak 2.1 taulan agertzen dira.

Mugarria	Data
Memoriaren entrega	2020/06/21
Aurkezpena prest izatea	2020/06/28
Lanaren defentsa	2020/06/29 - 2020/07/10

2.1 Taula: Mugarrien datak.

2.2.3 Denbora-plangintza

2.1 irudiko LDE diagramatik atera diren lan pakete bakoitzari dagokion aurreikusitako denbora 2.2 taulan agertzen dira.

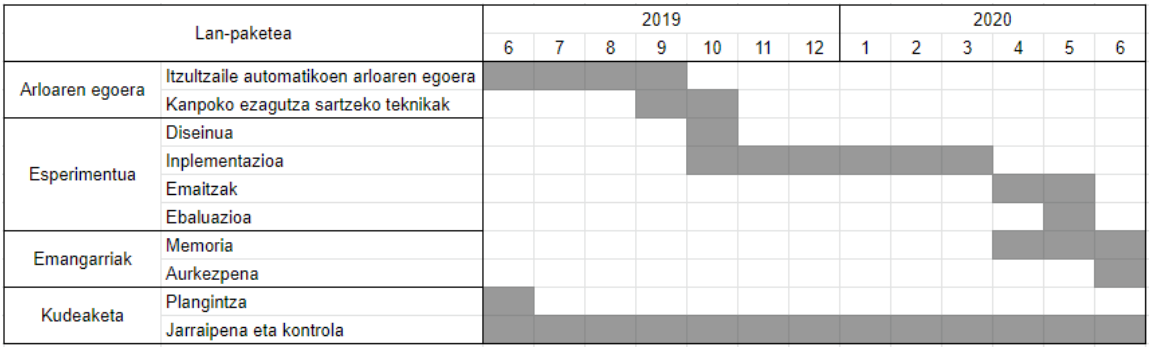
2.1 irudiko LDE diagramatik atera diren lan paketeak eta bakoitzari aurreikusitako denbora (2.2 taula) kontuan hartuta, Gantt diagrama 2.2 irudian ikus daiteke.

2.3 Lan metodologia

Atal honetan, lana garatzeko modua deskribatzen da. Interesatuak eta hauen eginbeharrak azaltzen dira, lanean zehar erabiltzen diren komunikazio bideak eta lana aurrera eramateko gertatzen diren bestelakoak.

Lan paketea	Aurreikusitako denbora
Arloaren egoera	60
Itzultzaile automatikoen arloaren egoera	45
Kanpoko ezagutza sartzeko teknikak	15
Esperimentua	135
Diseinua	10
Inplementazioa	110
Emaitzak	10
Ebaluazioa	5
Emangarriak	90
Memoria	80
Aurkezpena	10
Kudeaketa	15
Plangintza	5
Jarraipena eta kontrola	10
TOTALA	300

2.2 Taula: Lan pakete bakoitzaren aurreikusitako denbora.



2.2 Irudia: Lehenengo plangintzaren Gantt diagrama.

2.3.1 Interesatuak

Interesatuak lanean eragina duten pertsonak edota erakundeak dira, modu aktiboan naiz pasiboan parte hartzen dutenak. Lan honetako interesatuak hurrengoak dira:

- **Jesús Calleja.** Interesatu nagusia da proiektuaren egilea delako. Bere eginkizuna gradua amaitzeko helburu guztiak betetzen dituen proiektua amaitzea da. Gainera, itzulpen automatikoan jakin-mina du eta gai honetako arloaren egoeraren teknologiekin lan egiteko eta sakontzeko aukera izango du.
- **Gorka Labaka, Olatz Pérez de Viñaspre eta Ander Soraluze.** Interesatu hauek lanaren ikergaian interesa duten zuzendariak dira eta hauen eginkizuna lanaren jarraipena da, proiektuaren egileak izan ditzakeen zalantzak argituz, lanean zehar hartu behar diren erabakietan parte hartuz eta lana gidatuz.
- **Maite Oronoz.** Interesatu honek lanean erabiltzen den SNOMED CT ontologiarekin bateratuta dagoen FreeLing-Med programa du eta corpusa prozesatzeko lagungarria izango da.

2.3.2 Komunikazioak

Interesatuen arteko komunikazioak burutzeko hurrengo bi moduak planteatzen dira:

- **Bilerak.** Lanaren jarraipena egiteko, ikasleak zuzendariekin eskuarki bilerak burutuko ditu, astero baldintzek posiblea egiten badute, bai zuzendari guztiekin edo baten batekin, ordutegien eta haien denbora erabilgarritasunaren arabera. Bilerak posta elektronikoz edo aurreko bileraren amaieran hitzartuko dira. Bilerak fisikoki egiteaz gain, bideokonferentziaz ere egingo dira beste modu batean posible ez bada.
- **Posta elektronikoa.** Bileren beharra ez duten kontuak, bileren aldaketak edo premiazkoak diren gaiak komunikatzeko posta elektronikoa erabiliko da.

2.4 Arriskuen identifikazioa eta prebentzioa

Lanaren garapenean zehar ager daitezkeen arriskuei aurre egiteko, aurretik onuragarria da hauek identifikatzea eta izan dezaketen ondorio negatiboak leuntzeko prebentzio plana izatea.

Identifikatutako arriskuak eta bakoitzaren prebentzio plana jarraian agertzen dira:

- **Datuen galera.** Baliteke erabiltzen den makinen erabilgarri ez egotea eta datuak berreskuratzea posible ez izatea. Hau ekiditeko, zenbait segurtasun kopiak daude, bai *Google Drive* zerbitzuan, bai IXA taldeak dituen zerbitzarietan, bai ikasleak duen makinan eta kanpoko memorian.

- **Itzultzaile automatikoa entrenatzeko denbora luzeegia izatea.** Ikasketa sakoneko ereduek denbora asko behar dute entrenatzeko bere parametro kopuruagatik eta behar duten datu kopuruagatik. Horregatik, IXA taldeko GPUak erabiltzen saiatuko da edo hauek erabilgarri ez badaude, *Google Colab* zerbitzua erabiliko da, Googlek eskuragarri uzten dituen GPUak erabiliz. Horretaz gain, eredua entrenatzen den bitartean, denbora ez galtzeko, lanarekin lotuta dauden beste atazak burutu daitezke, hala nola, memoriaren idazketa.

- **Egindako plangintzan desbiderapen handiegiak egotea.** Gerta daiteke plangintzan egin diren aurreikuspenak motz geratzea eta lan pakete batzuekin eman behar den denbora hasieran pentsatutakoa baino handiagoa izatea edota eskuragarri izandako denbora pentsatutakoa baino gutxiagoa izatea. Hau gertatuz gero, beste atazen denbora murriztu daiteke, hauen emaitzen kalitatea kaltetuz, edo posiblea bada, iraileko deialdian lana aurkeztea, gertatu diren zailtasunak konpontzeko denbora gehiago izateko.

2.5 Proiektuaren bideragarritasuna

Lanaren plangintza behin ikusita, honen bideragarritasuna bermatzen duten puntuak jarraian garatzen dira:

- **Lanaren kostu ekonomikoa.** Lanaren helburuak betetzeko behar diren baliabideak doakoak direla bermatzen da.

- **Lanaren denbora kostua.** Lana burutzeko denbora dagoela bermatzen da, plangintzan justifikatu den bezala.

- **Lanaren funtzionamendua.** Lanak erabiltzen dituen baliabideak modu egokian dihardutela bermatzen da.

- **Lanaren jarraipena.** Lanaren garapen egokia egoteko, zuzendariekin dauden komunikazioak egokiak direla bermatzen da.

3. KAPITULUA

Aurrekariak

Atal honetan, lanaren atal teknikoaren oinarriak azaltzen dira. Proiektuan hartutako diseinu erabakiak ulertzeko, beharrezkoa da jakitea zeintzuk izan diren orain arte erabili diren teknologiak. Hasteko, itzultzaile automatikoen arloko sarrera emango da, zertan datzan azalduz. Ondoren, itzultzaile automatikoak garatzeko arkitektura berrienetan sakonduko da. Arkitektura bakoitzaren funtzionamendua azalduko da, alde indartsuekin eta ahuleziekin batera. Amaitzeko, azaldutako arkitektura batzuetan kanpoko ezagutza sartzeko erak aztertzen dira, beste ikerlariek egin dituzten aurkikuntzetaz ikasteko eta lan honetan egingo den proposamena ahalik eta hobekien garatzeko.

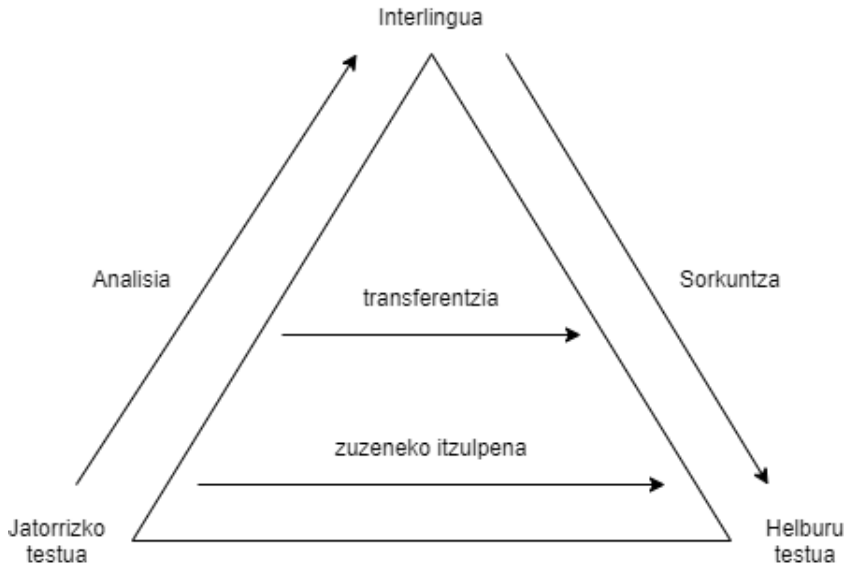
3.1 Itzulpen automatikoa

Itzulpen automatikoa *software* bitartez hizkuntza batetik bestera testu bat itzultzeko prozesua ikertzen duen hizkuntzaren prozesamenduaren arlo bat da. Urteetan zehar, aurrerago azalduko denez, paradigma desberdinak erabili dira prozesu hau lortzeko [Hutchins and Somers, 1992]. Itzulpen automatikoaren bidea erregeletan oinarritutako itzulpen automatikoarekin hasten da 1950. hamarkadan. Izen berak adierazten duen bezala, erregelak erabiltzen dira itzulpena burutzeko. Ordenagailuen konputazio ahalmena gero eta handiagoa bihurtzearekin handituz eta kostua merketuz, 1980. hamarkadaren amaieran, itzulpen automatiko estatistikoa erabiltzen duten modeloak agertzen hasten dira. Modelo hauek, erregelekin ez bezala, jatorrizko eta helburu hizkuntzaren arteko probabilitate dis-

tribuzioa ikasten dute. Orainaldian, sare neuronalak erabiltzen dituzten itzultzaileak dira emaitza onenak ematen ari direnak.

3.1.1 Erregeletan oinarritutako itzulpen automatikoa

Erregeletan oinarritutako itzulpen automatikoa, *rule-based machine translation* (RBMT) ingelesez, aditu batek eskuz idatzitako erregelak oinarri erabiltzen ditu itzulpenak burutzeko.



3.1 Irudia: RBMT paradigmaren hiru teknika desberdinen Vauquois triangelua.

3.1 irudian agertzen den bezala, hiru teknika desberdin daude RBMT paradigmaren barruan itzulpen automatikoa egiteko, bakoitzak sakonera handiagoarekin.

- **Zuzeneko itzulpena.** Teknika honetan hitz bakoitzaren itzulpena egiten da hiztegi batekin egingo balitz bezala. Hasierako analisi morfologikoa, hau da, hitz bakoitzaren kategoria gramatikalaren (aditzondoa, izena, izenlaguna...) identifikazioa eta lematizazioa, hitzen lemen identifikazioa (adb. *etxeko*, *etxearen* eta *etxean* hitzen lema *etxe* da). Teknika honen arazo nagusia hizkuntzen arteko desberdintasun linguistikoen arabera motz geratzen zela.
- **Transferentzia.** Teknika honekin, analisi morfologikoa eta lematizazioa egin ondoren, itzuliko denaren analisi sintaktikoa (aditza, subjektua, objektu zuzena...) egiten da. Pauso honetan geratuz gero, transferentzia sintaktikoa egingo litzateke. Honekin

batera, analisi semantikoa (hitzen adieren desanbiguazioa) eginez gero, transferentzia semantikoa egiten dela esan daiteke. Analisiaren ondoren transferentzia (helburu hizkuntzako zuhaitz sintaktikoa lortzea) eta sorkuntza (helburu hizkuntzako testua lortzea) faseak datoz. Beste modu batean esanda, jatorrizko testuaren analisitik ateratako informazioa helburu hizkuntzara itzultzeko erabiltzen da bitarteko transferentzia programan. Zuzeneko itzulpenarekin kontrastatuz, analisian lortutako informazio linguistiko sakonagoak transferentzia edo itzulpen fasea errazten du.

- **Interlingua**. Teknika honek jatorrizko testutik bitarteko errepresentazioa lortzen du, *interlingua* deritzona, eta helburu hizkuntzako testua sortzen du. Errepresentazio hau jatorrizko testuaren esanahiaren errepresentazioa da eta hizkuntzekiko independentea da. Era berean, analisirako eta sorkuntzarako programak independenteak dira, hizkuntza bikote desberdinen itzulpena erraztuz.

3.1.2 Itzulpen automatiko estatistikoa

Itzulpen automatiko estatistikoa, *statistical machine translation* (SMT) ingelesez, jatorrizko eta helburu hizkuntzan dauden corpus paraleloen probabilitate banaketan oinarritzen da [Kirchhoff and Yang, 2005]. Itzulpen automatikoa egiteko modu berri hau lehenengo aldiz [Weaver, 1955]-ek planteatu bazuen ere, [Brown et al., 1990]-ek geroago praktikan jarri zuen konputazio ahalmena handitzeari esker. Paradigma honen ideia nagusia hurrengoa da: demagun ingelesetik gaztelerara esaldi bat itzuli nahi dugula. Hiztegi bikote hauen corpus paralelo asko existitzen direnez, hauek erabiliko dira emandako gaztelaniazko esaldi bat ingelesezko esaldi bati dagokion probabilitatea lortzeko. Eragiketa gaztelerazko beste esaldi askorekin egiten da eta probabilitate handiena duen esaldia hartzen da. Prozesua berdina da hitz bat, esaldi bat edo testu batekin. Gaztelerazko itzulpenaren probabilitatea Bayes-en teoremaren formularekin (3.1 ekuazioa) kalkulatzen da.

$$P(S|O) = \frac{P(O|S)P(S)}{P(O)} \tag{3.1}$$

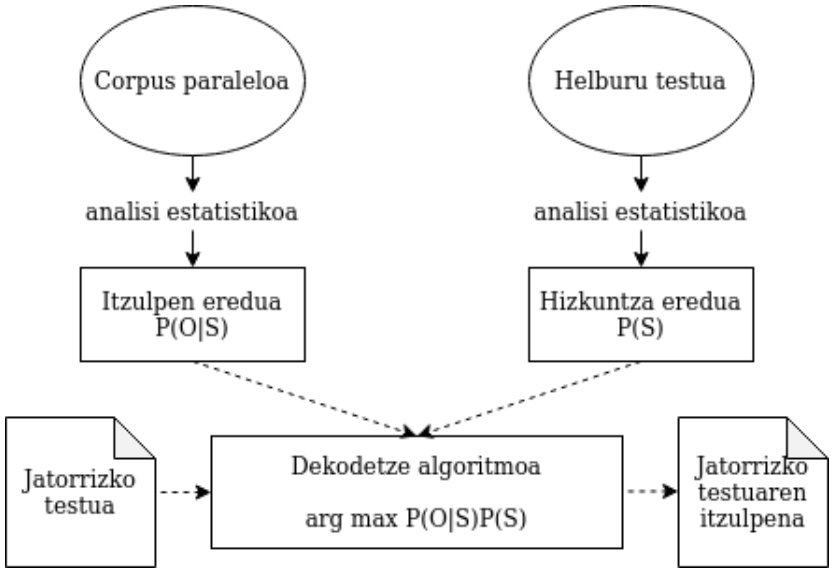
Ekuazio honen esanahia hurrengoa da: S egoera baten probabilitatea O gertakizun bat emanda, $P(S|O)$, gertakizunaren probabilitatea egoera bat emanda, $P(O|S)$, bider egoeraren probabilitatea, $P(S)$, zati gertakizunaren probabilitatea, $P(O)$, da.

Kasu honetan, ingelesezko esaldiaren probabilitatea, $P(O)$, gaztelaniazko esaldi guztietarako, egoerarako, berdina izango denez, bakarrik $P(O|S)P(S)$ kontuan hartu behar da.

Hortaz, sistemak eragiketa hori maximizatzen duen g^* multzoaren esaldi guztien arteko gaztelerazko $\tilde{g}$ esaldia bueltatuko du ingelesezko i esaldiaren itzulpena bezala, 3.2 ekuazioan agertzen den moduan.

$$\tilde{g} = \arg\max_{g \in g^*} p(g|i) = \arg\max_{g \in g^*} p(i|g)p(g) \tag{3.2}$$

Beraz, $\tilde{g}$ kalkulatzeko bi elementu nagusi ulertu behar ditugu: $P(i|g)$, ingelesezko esaldi bat gaztelerazkoaren itzulpena izatearen probabilitatea, eta $P(g)$, gaztelerazko esaldi bat erabilita izatearen probabilitatea (*a priori* ezagutza), edo, beste modu batean adierazita, itzulpen eredua eta hizkuntza eredua, hurrenez hurren.



3.2 Irudia: SMT oinarrizko arkitekturaren diagrama.

SMT arkitekturek hiru elementu nagusi dauzkate: hizkuntza eredua, itzulpen eredua eta deskodetze prozesua, 3.2 irudian agertzen den bezala.

- **Hizkuntza eredua.** Elementu honek, $P(g)$, gaztelerazko esaldi bat zuzena dela bermatzen du. *A priorizko* probabilitate hauek kalkulatzeko erabilienak n-gramak dira, n hitzetako parez pareko sekuentziak. “Me gustan los perros” esaldian, trigrama edo hiru hitzetako n-grama baten adibidea “me gustan los” izango litzateke. Metodo honen bitartez, hizkuntzaren gramatika jakin gabe, ereduak hitzen ordena eta erabilera ikasten du. Hala ere, luzera handiagoko n-gramak ere erabili daitezke, baina konputazio kostua handituz. Eredua ez da perfektua izan behar; gaztelera zuzenak okerra baino probabilitate handiagoa duela bakarrik bermatu behar du kasu gehienetan.

- **Itzulpen eredua.** Itzulpen eredua, $P(i|g)$, ingelesezko esaldi bat gaztelerazko esaldi baten itzulpena den probabilitatea kalkulatzen du. Atal hau burutzeko, ereduak corpus paraleloetatik ikasten du. Aurreko kasua hartuta, sistemak gaztelerazko hitzak zein ingelesezko hitzen itzulpenak diren zehaztu behar du. Horretarako, sistemak ingelesezko hitz bat duten esaldien itzulpenen zein hitz agertzen diren maizago begiratuko du. Ereduak amaieran estatistikoki hitz bat bestearen itzulpena dela adierazteko gai da. Eredu honek zailtasun batzuk corpusen arteko hitzen lerrokatzea eta *emankortasun* kontzeptuak dira, non hizkuntza bateko hitz baten itzulpena hitz bat baino gehiagoz eginda dagoen.

Hizkuntza eredua eta itzulpen eredua batuta hizkuntza baten esaldiaren itzulpena lor dezakegu. Hala ere, itzulpen hautagai asko eta hizkuntza partikular baten konplikazio asko daudenez, bilaketa optimo bat egitea posiblea bada ere, normalean azkarrago eta emaitza onak ematen dituen bilaketa azpioptimoak erabiltzen dira. Prozesu honi **deskodetzea** deritzo. Erabiltzen diren bilaketa ohikoenak eta erabilienak *greedy search* edo bilaketa jalea eta *beam search* dira. Lehenengoak itzulpenaren hurbil dauden aldaerak aztertzen ditu, jatorrizkotik aldaketa minimoak dituena, eta bertsio berri bat erabiltzen du probabilitatea hobetuz gero. Bilaketa honen azken iterazioak kontuan hartuz, maximo lokalak saihestu daitezke konputazio kostu oso txikiarekin. Bigarrenarekin, berriz, luzera handiko esaldiekin batez ere konputazio kostua asko igotzen da. *Beam search* bitartez, hiperparametro baten arabera, hasierako hautagai batzuen aldaerak aztertzen dira eta horietatik ziurtasun gehien edo probabilitate handien eskaintzen dituzten k aldaerak gehiago aztertuko dira. Prozesu hau aldaera bakoitarekin errepikatzen da, zuhaitz bezalako egitura bat eginez.

3.1.3 Itzulpen automatiko neuronala

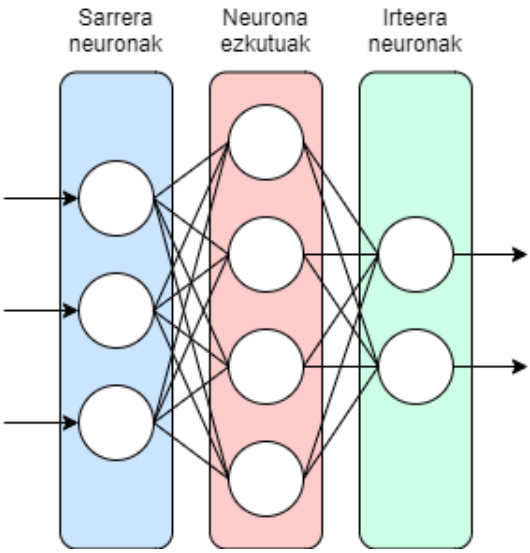
Itzulpen automatiko neuronala, *neural machine translation* (NMT) ingelesez, itzultzeko sare neuronalak erabiltzen dituen paradigma da. Aurrekoak ez bezala, paradigma honek pauso guztiak eredu batean integratzen ditu, *end-to-end* deritzona. SMT ereduekin gertatu zen bezala, aspaldi sare neuronalen ideiak proposatu ziren, baina konputazio kostua dela eta ezinezkoa izan zen ataza konplexuak burutzeko eredu hauen benetako ahalmena ikustea orain dela gutxi arte. Sare neuronal artifizial baten lehenengo ideiak [McCulloch and Pitts, 1943] eman zituzten eta urteak pasa ahala hobekuntzak egin dira, gaur egungo egoerara heldu arte. Itzulpen automatikoaren atazarako, SMT ereduen ondorengoak izan ziren. Lehen aldiz, itzulpen automatikorako [Kalchbrenner and Blunsom, 2013] sare

neuronal bat erabiltzea proposatu zuten, *word embeddingen* (3.3 atala) etorkizun handiko emaitzak aprobetxatuz. Geroago, arkitektura berriak agertu dira bakoitzak aurrekoaren arazoak konponduz edo emaitza hobeak ematen dituzten teknologiak erabiliz. Gaur egun, emaitza hoberena ematen duen arkitektura Transformerra da, 3.2.3 atalean azaltzen dena.

3.2 Itzulpen automatiko neuronalen arkitekturak

Sare neuronalak burmuinen funtzionamenduan inspiratuta dauden arkitekturak dira. Bi elementuz osatuta daude: neuronak eta haien arteko konexioak. Konexio hauek neurona baten irteera bestearen sarreraren parte direla errepresentatzen dute. Neurona baten irteera kalkulatzeko, aurretik konektatuta dauden neuronen irteerekin eta haien konexioen pisuarekin edo garrantziarekin kalkulatzen da. Azken hauek dira ereduak entrenatzean aldatzen diren parametroak.

Ereduak hiru motako neurona motak ditu: sarrerakoak, ezkutuak eta irteerakoak. Sarrerako neuronek kanpoko informazioa izango dute sarrera bezala. Neurona ezkutuak sarrerako neuronekin edo beste neurona ezkutuekin konektatuta egongo dira. Normalean, neurona ezkutuko zenbait geruza erabiltzen dira. Azkenik, irteera neuronek ereduaren emaitza bueltatuko dute. Adibidez, sailkapen-ataza batean irteera neurona adina klase egongo da eta ereduak bueltatuko duena klase bakoitzaren probabilitatea izango da, sarreraren araberakoa izango dena. 3.3 irudian oinarrizko sare neuronal baten adibidea ikus daiteke.



3.3 Irudia: Oinarrizko sare neuronal bat, geruza motekin nabarmenduta.

Sare neuronal baten konfigurazioak bi fase nagusi ditu: entrenamendu fasea eta proba fasea. Fase bakoitzerako bi datu-multzo edo *dataset* desberdin erabiltzen dira: bata entrenamendurako eta bestea eredua ebaluatzeko.

Entrenamendu fasearen helburua ereduaren parametro guztiak konfiguratzea da. Horretarako, *gradient descent* teknika erabiltzen da. *Datasetean* sarrera bakoitzerako irteera jakin bat dauka. Ereduan sarrera sartzen da eta ereduak momentuan dituen parametroekin irteera bat lortzen dute. Honi aurrerazko edo *forward* fasea deritzo. Irteera hori datasetean agertzen zenarekin konparatzen da eta *loss function* edo errore funtzio baten bitartez kalkulatzen da. Ohikoenak *Mean Square Error* (MSE) (3.3 ekuazioa) eta *Cross Entropy* (3.4 ekuazioa) dira, non $\hat{y}$ ereduak iragartzen duen erantzuna den eta y benetako erantzuna den. Behin errorea kalkulatu dela *backpropagationari* hasiera ematen zaio. Irteera geruzaren errorea errore-funtzioaren deribatua parametro bakoitzarekiko kalkulatzen da. Deribatuaren kontrako noranzkoan parametroa eguneratzen da. Berdina egiten da hurrengo geruzekin, *forward* fasearen kontrako ordenean. Behin parametro guztiak eguneratzen direla, prozesua berriro hasten da entrenamenduko datu guztiekin.

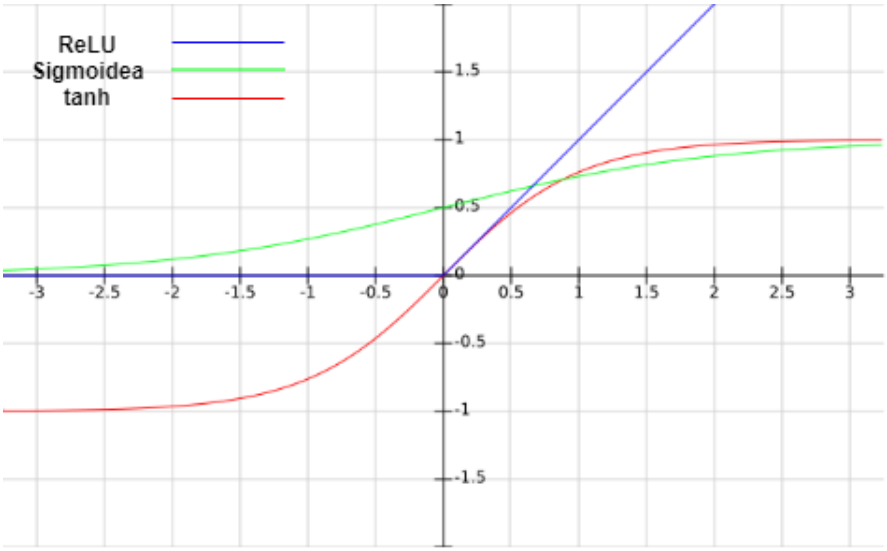
$$MSE = \frac{1}{n}\sum(y-\hat{y})^2 \tag{3.3}$$

$$Cross\ Entropy = -\sum(y * log(\hat{y})) \tag{3.4}$$

Proba fasearen helburua eredua ebaluatzea da. Horretarako, lehen esan bezala, ereduak entrenatzeko erabili ez dituen datuak erabili behar dira. Fase honetan aztertu daitekeen elementu garrantzitsuena orokortzeko gaitasuna da. Eredua ikusi ez duen datuekin emaitza onak emateko gai izan behar da.

Oinarrizko adibidea *Multilayer Perceptron* (MLP) da (3.3 irudia). Sare hau *Feed Forward Neural Network* (FFNN) motakoa da, hau da, neuronen konexioak ez dira inoiz aurreko neuronekin egiten. Sinpleena geruza edo *layer* batekoa da, izan ere, geruza bateko MLPa erregresio lineala baino ez da. Atazaren beharren arabera eta ereduaren entrenamenduari begira, geruza gehiago gehitzen zaizkio ezlineartasun gehiago sartzeko. Ezlineartasun hau aktibazio funtzioen bitartez sartzen da. Neurona baten irteera kalkulatzean ez dira bakarrik beste neuronen irteerak eta haien pisuak erabiltzen. Normalean, aktibazio funtzio ez linealak erabiltzen dira ereduaren konplexutasuna handitzeko. Erabiltzen diren ohikoenak *Rectified Linear Unit* (ReLU), sigmoidea eta *tanh* dira. Funtzio bakoitzaren irudiaren heina desberdina da eta, beraz, atazaren balioen arabera batzuk besteak baino hobeak dira,

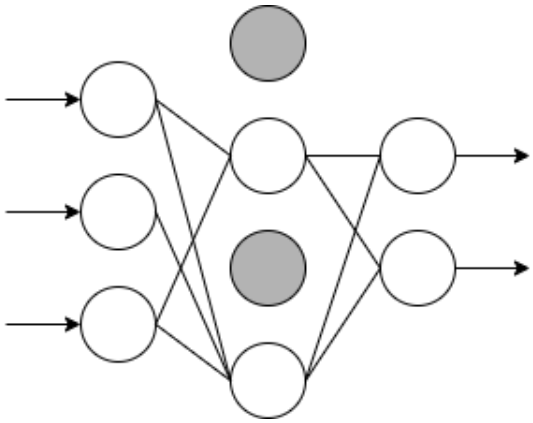
3.4 irudian ikusten den moduan. Adibidez, ReLUren kasuan, zenbaki negatiboek balio berdina izango dute. Beharbada, sareak informazio esanguratsua eman nahi du balio negatibo handiekin, baina ReLUk 0-ra jartzen ditu eta ereduaren entrenamendua kaltetzen du.



3.4 Irudia: ReLU, sigmoidea eta *tanh* funtzioak.

Atazaren edo beharren arabera, neurona kopurua eta konexioak aldatzen dira, arkitektura deritzonak. Horrela, lehen esan bezala, konplexutasuna gehitzen zaio ereduari eta emaitza hobetzeko ahalmena ematen zaio. Hala ere, kontuan hartu behar da atazaren zailtasunaren proportzio berdinean konplexutasuna gehitu behar dela. Ataza zail baterako sare neuronal sinple bat erabiltzen bada, eredua ez da ikasteko gai izango. Honi *underfitting* deritzo. Era berean, eredua atazarentzat konplexuegia bada, eredua gehiegi ikasiko luke *overfitting*a gertatuz. Hau gertatzen da eredu bat emaitza onak entrenamendurako datuekin ematen dituenean, baina ez hainbeste proba edo *test* kasuetarako. Hau kaltegarria da sareak orokortzeko gaitasuna galtzen duelako. *Overfitting*a ekiditeko, erregularizazio teknikak erabiltzen dira. Hauek ereduaren konplexutasuna zigortzen dute eta eredua entrenamendu datuetara gehiegi doitzea ekiditzen dute, orokortzeko gaitasuna lortuz. Kasurik ohikoena *dropout* da.

Dropout teknikaren bitartez, entrenamendu faseko pauso bakoitzean ausaz aukeratutako neuronak desaktibatzen dira [Srivastava et al., 2014] (3.5 irudia). Horrela, ereduak neurona horiek erabili gabe ikasiko du eta orokortze handiagoa lortuko da. Desaktibatuta dauden neuronen sarrerako konexioak eta irteerakoak ez dira erabiliko eta horien parametroak pauso horretan ez dira erabiliko ezta eguneratuko.



3.5 Irudia: *Dropoutaren* adibidea geruza batean.

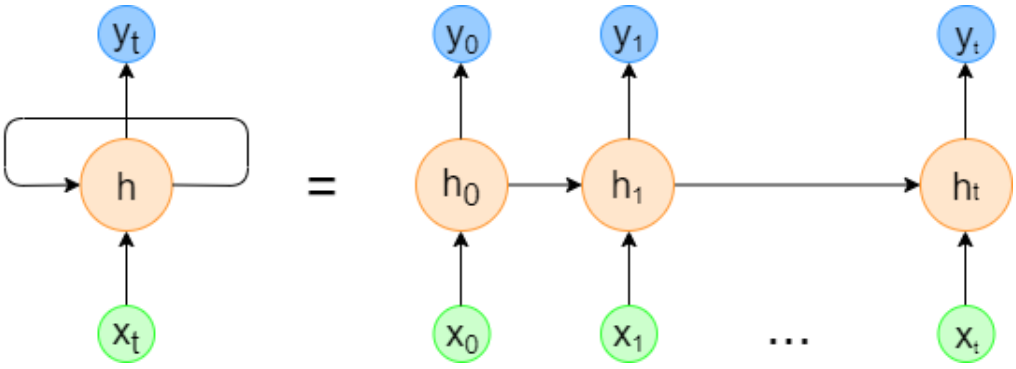
NMT atazan erabili diren arkitektura hoberenak Long-Short Term Memory (LSTM), Encoder-Decoder eta Transformer dira.

3.2.1 Sare errekurrenteak

Sare errekurrenteak, *recurrent neural network* (RNN) ingelesez, hurrengo pausoetan kalkulatzen diren neuronen irteerak aurreko pausoetako neuronen sarrerari konektatuta daudenean aurkitzen dira. Sare hauek oso erabilgarriak dira datu sekuentzialekin; lan honetan, hizkuntza naturalarekin.

Eredu honen adibide sinpleena 3.6 irudian ikus dezakegu, bere forma destolestuarekin batera. Izan ere, sare errekurrente bat oso sakona den FFNNa baino ez da, baina sakonera aldakorra eskaintzen du. Hortik datu sekuentzialekin hain ona izatea dator. Sareak x_t sarrera dauka eta y_t irteera ematen du. Tartean, h_t aurkitzen da. Honi *hidden state* edo egoera ezkutua deritzo. Egoera ezkutuak aurreko sarreren informazioa gordetzen du eta pauso bakoitzaren sarrerarekin eguneratzen da. FFNNekin bezala, konexio bakoitzak ere bere pisuak dauzka.

Sare hauek bi arazo nagusi dituzte: *gradient vanishing* eta *gradient exploding*. Lehen azaldu da ereduak parametroen balioak ikasteko *gradient descent* erabiltzen duela. RNNak sarrera oso luze baterako *backpropagation* pausoan kateatuta dauden gradienteak asko dira. Beraz, bi gauza gerta daitezke: gradientea oso txikia bihurtzea eta parametroak ia ez aldatzea edo kontrakoa; gradientea gero eta handiagoa egitea eta parametroen balioa gehiegi aldatzea. Arazo hauek ereduaren entrenamenduaren gelditzea edo ezegonkortzea eragiten dute. Edozein kasutan, ereduarentzat ezinezkoa izango da parametroak modu egokian



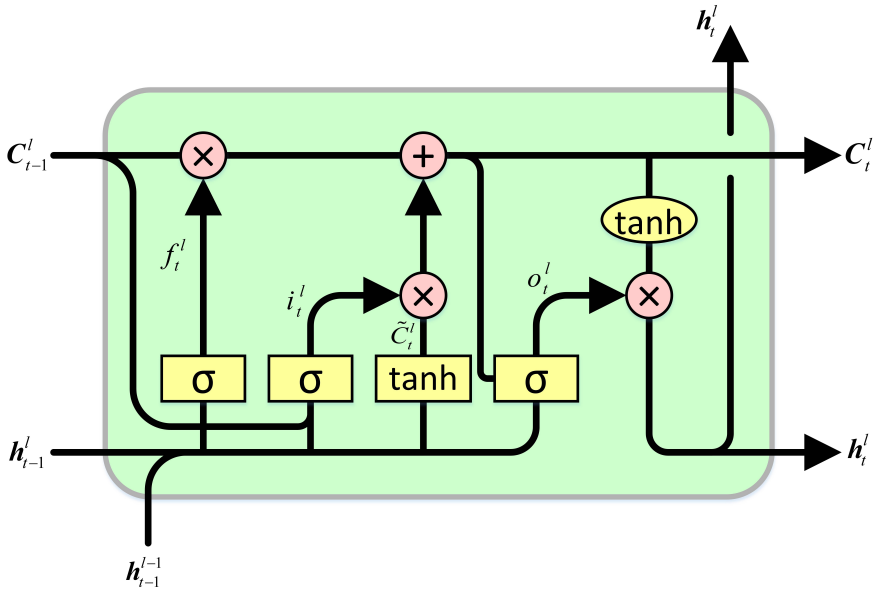
3.6 Irudia: RNN simplea, bere forma destolestuarekin batera.

ikastea.

Arazo hauek ekiditeko ateetan edo *gate*tan oinarritzen den LSTM arkitektura proposatu zen [Hochreiter and Schmidhuber, 1997]. Ateek zenbat informazio pasatzen den zehazten dute. Gainera, *cell state* edo zelularen egoera (c_t) gehitzen da egoera ezkutuarekin (h_t) batera. Zelularen egoera memoria bezala ikus daiteke. RNNetan egoera ezkutuak irteera eta sekuentziaren informazioa da aldi berean. LSTMetan, berriz, laguntza ematen zaio memoria elementu bat gehituz. Pauso bakoitzean, ereduak sarreraren eta egoera ezkutuaren informazioarekin zer egin aukeratuko du: aurreko informazioa ahaztu *forget gate*aren edo ahazte atearen bitartez (f_t) eta informazio berria ikasi *input gate*arekin edo sarrera atearekin (i_t). Neuronaren irteera *output gate*aren edo irteera atearen (o_t) eta zelularen egoeraren konbinazioarekin egiten da. Horrela, LSTMaren ideia orokorra da memoria elementu bat dagoela sekuentziaren informazioa gordetzeko edo ahazteko egoera ezkutuaren laguntzaz. Egoera ezkutua memoriaren, pauso horren sarreraren eta aurreko egoera ezkutuaren arabera eguneratzen da, bakoitzak informazio desberdina eskainiz. LSTM zelula baten egitura 3.7 irudian ikus daiteke.

3.2.2 Encoder-Decoder eta sekuentziatik sekuentziarako arkitektura

Encoder-Decoder arkitekturak 3.1.1 atalean azaltzen den interlinguaren ideia berdina du, baina bektoreak erabiltzen dira tarteko hizkuntza bat erabili beharrean. Arkitekturak bi zati nagusi ditu: kodetzailea eta deskodetzailea. Kodetzaileak sarrerako datuetatik tarteko egoerako errepresentazioa lortzen du. Tarteko errepresentazio horrek sarreraren ezaugarri latenteak gordeko ditu bektore batean, testuinguru edo mapaketa bezala erabiliko dena eta deskodetzaileak irteera aukeratuko du sarrera eta tarteko errepresentazioa kontuan hartuz.



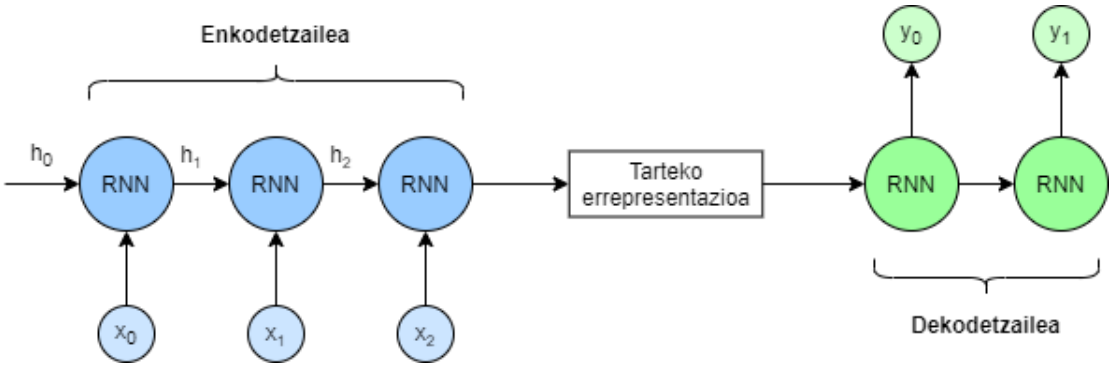
3.7 Irudia: LSTM zelularen egitura.

Atazaren arabera, arkitekturaren bi zatietan komeni den sare neuronala inplementatzen da.

Encoder-Decoder arkitekturaren kasu partikularra sekuentziatik sekuentziarakoa (*seq2seq*) da. Lehenengoarekin sarrera bakarra erabiltzen da eta irteera bakarra lortzen da; seq2seq-rekin, ordea, sarrera bezala sekuentzia bat ematen zaio eta irteera sekuentzia bat izango da, NMT atazarako erabilgarria dena. Horretarako, RNNak erabiltzen dira, lehen esan den moduan, sekuentziak prozesatzeko erabilgarriak direnak. RNN hauek LSTMak [Sutskever et al., 2014] edo Gated Recurrent Unit (GRU) ere izan daitezke [Cho et al., 2014]. Azken hau LSTMaren aldaketa baino ez da. Arkitektura honen errepresentazio orokorra 3.8 irudian ikus daiteke.

Arkitektura honek daukan arazo nagusi bat RNNak duten berdina da: epe luzerako menpekotasunak. Esaldi motzetan arkitekturak emaitza onak ematen ditu, baina esaldi luzeagoak erabili ahala emaitzen kalitatea gero eta baxuagoa da. Horretarako, *atentzioa* deritzon teknologia erabiltzen da.

Atentzioak sekuentziaren zati garrantzitsuenak identifikatzen ditu eta testuinguru bektore bat lortzen du iragarpenak egiteko [Luong et al., 2015]. Kodetzaileak pauso bakoitzean zehar egoera ezkutu desberdin bat du, puntu horretara arte sekuentziaren informazioa gordetzen duena. Atentzioaren bitartez, egoera ezkutu bakoitzaren garrantzia deskodetzailearen uneko egoera ezkutuko biderketa eskalarrarekin kalkulatzen da. Ondoren, pisu



3.8 Irudia: *Seq2seq* arkitektura.

horiei 0 eta 1 arteko balioak esleitzen zaizkie *softmax* funtzioa erabiliz (3.5 ekuazioa), normalizatzeko funtzio esponentziala dena. Atentzioaren testuinguru bektorea lortzeko, kodetzailearen egoera ezkutu bakoitza bere garrantziarekin biderkatzen da. Amaitzeko, atentzioaren testuinguru bektorea eta uneko deskodetzailearen egoera ezkutua erabiliz atentzio bektorea lortzen da, irteera erabakitzeko erabiliko dena. 3.9 irudian azaldutakoa ikus daiteke.

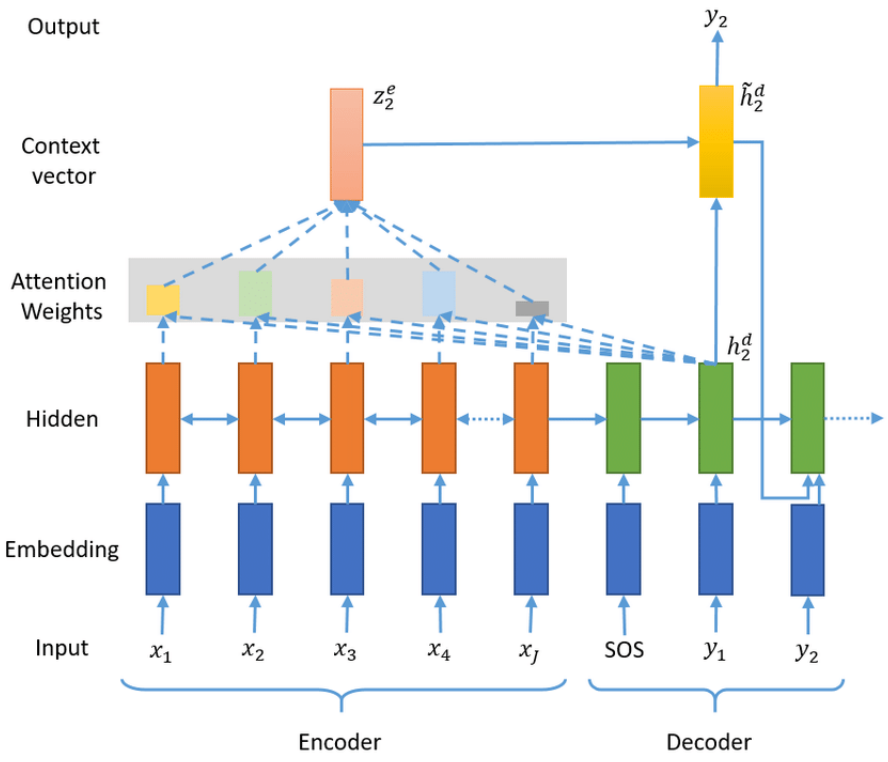
$$Softmax\,(a) = \frac{e^{a_j}}{\sum_j e^{a_j}} \tag{3.5}$$

Kodetzaileko egoera ezkutuen aukeraketa egiteko bi modu daude: atentzio globala eta lokala. Atentzio globalean sekuentzia osoko egoera ezkutu guztiak kontuan hartzen dira; lokalekoan, ostera, bakarrik horietako batzuk, hiperparametro baten araberakoa, hartzen dira atentzio bektorea kalkulatzeko.

Beraz, atentzioaren helburua ez da sekuentzia osoari garrantzia ematea, *seq2seq* normalek duten tarteko errepresentazio horrekin bezala, baizik eta sekuentziako atal jakin batzuei arreta jartzea, garrantzitsuenak direnak, eta horien informazioa erabiltzea iragarpen egokiak egiteko.

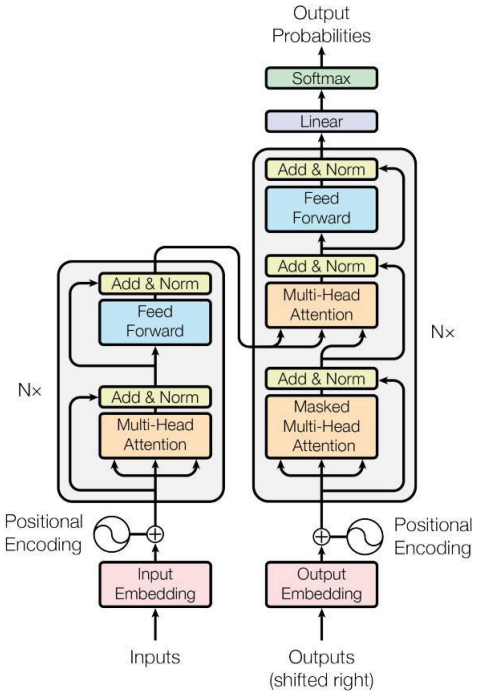
3.2.3 Transformer

Seq2seq arkitekturan oinarrituta, [Vaswani et al., 2017]-ek Transformer arkitektura proposatu zuen (3.10 irudia). Oinarria berdina da, baina kodetzailean RNN sare bat erabili beharrean, arkitektura honek *Multiheaded attention* deritzon arkitektura erabiltzen du. Honen bitartez, ereduak ezaugarri desberdinak eta patroi gehiago identifikatzeko ahal-



3.9 Irudia: *Seq2seq* arkitektura atentzioarekin. [Shi et al., 2018]

mena lortzen du. Horretaz gain, Transformerrak RNNak erabiltzen ez dituenez, hauek sekuentzia luzeetatik deribatzen dituzten arazoak ez dauzkate.



3.10 Irudia: Transformer arkitektura. [Vaswani et al., 2017]

RNNekin sententziaren *i*. posizioa prozesatzeko, *i-1*.a prozesatu behar da halabeharrez. Transformerrak sare neuronal mota hori ez duenez erabiltzen, sekuentzia osoa aldi berean prozesatu daiteke, ikasketa azkartuz eta paralelizatzeko aukera eskainiz.

Transformerrak Encoder-Decoder arkitekturan oinarrituta dago. Zati bakoitzak *n* geruzaz osatuta dago. Kodetzaile eta deskodetzailearen berdintsuak dira, baina gero azalduko den moduan deskodetzaileak kodetzailearen pisuak erabiltzen ditu testuinguru bezala.

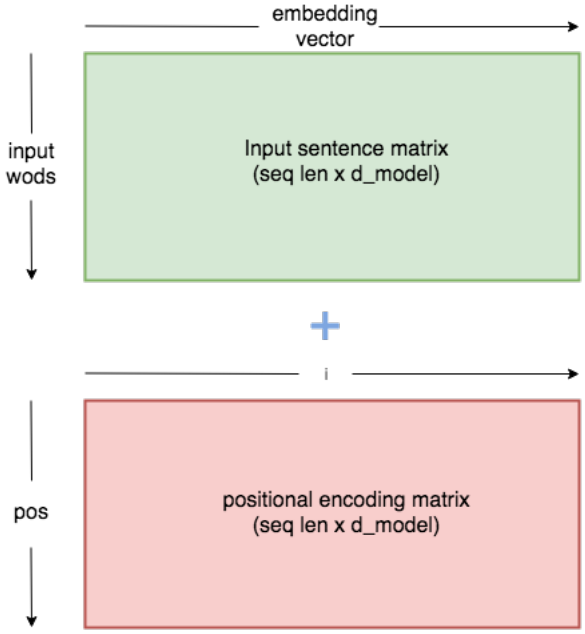
Hasteko, sarrerako datuak *embedding* geruza batetik pasatzen da. NMT atazan, hitzak errepresentazio jarrai batera pasatzen dira (3.3 atala). Horrela, ereduari ez zaio *string* hori pasatzen, baizik eta hitz hori errepresentatzen duen zenbakizko bektorea. Geruza hori entrenatzeko datasetean erabiltzen den hitz bakoitzaren bektorea gordetzen du matrize batean. Hitz bat geruzan sartzerakoan, hitz horri dagokion indizea lortzen da eta matrizeko indize horretako bektorea bueltatzen du.

RNNetan, sekuentziako elementuak bata bestearen atzetik prozesatzen zirenez, ereduak elementuen arteko ordenaren informazio inplizitua zeukan. Transformerrean, elementuak

paraleloki prozesatu daitezke, baina ezer gabe ereduarentzat ezinezkoa izango litzateke sekuentziaren barruan elementuaren posizioa zehaztea. Hori dela eta, nolabaiteko informazioa gehitu behar zaio *embedding* bakoitzari. Horretarako, *positional encoding* geruza erabiltzen da. Geruza honek 3.6 eta 3.7 ekuazioak erabiltzen ditu eta hauek gehitzen ditu embeddingetan ereduak esaldien barruan hitzek dauzkaten orden erlatiboak ikasteko. *pos* sekuentzia batean hitzaren posizioari dagokio eta *i* hitzaren embeddingaren posizioari dagokio. Sortzen den matrizea 3.11 irudian agertzen bi matrizeen arteko baturarena da.

$$PE(pos, 2i) = sin\left(\frac{pos}{10000^{2i/d_{model}}}\right) \tag{3.6}$$

$$PE(pos, 2i+1) = cos\left(\frac{pos}{10000^{2i/d_{model}}}\right) \tag{3.7}$$



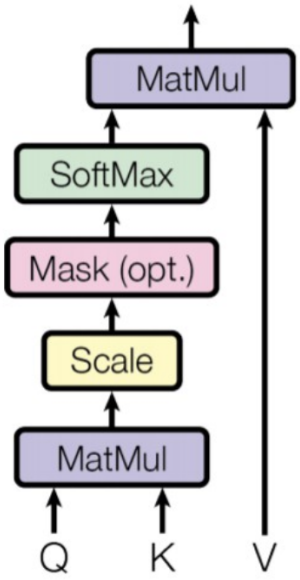
3.11 Irudia: *Positional encoding* geruza.

Behin hitzen embeddingak prest daudela, hauek kodetzailean sartzen dira. Kodetzaileak *n* geruza dauzka. Geruza guztiak berdinak dira, baina bakoitzak bere pisu propioak dauzka. Lehenengo elementua *Multiheaded Attention* edo *Self-Attention* azpigeruza da. Azpigeruza hau arkitekturaren atal nagusia da, bertan sekuentzien elementuen (embeddingen) arteko dependentziak identifikatzen dira eta informazio hori erabiltzen da sekuentziaren egitura ulertzeko. Ideia nagusia 3.2.2 ataleko atentzioarena da, baina zenbait aldaketa

dauzka. Geruza hiru matrizez osatuta dago: *query* (Q), *key* (K) eta *value* (V). Matrize hauek berdinak izango dira eta sekuentzietaz osatuta egongo dira. Matrize hauei transformazio linealak aplikatuko zaizkie parametroen matrizeekin biderkatuz (3.8 ekuazioa, non X sarrerako matrizea den, W geruzaren parametroen matrizea eta Y bi aurreko matrizeen arteko biderketaren emaitza) eta *scaled dot product attention* deritzona aplikatuko da matrize hauen gainean (3.9 ekuazioa, non d_k key matrizearen dimentsioa den).

$$Y = XW \tag{3.8}$$

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{3.9}$$

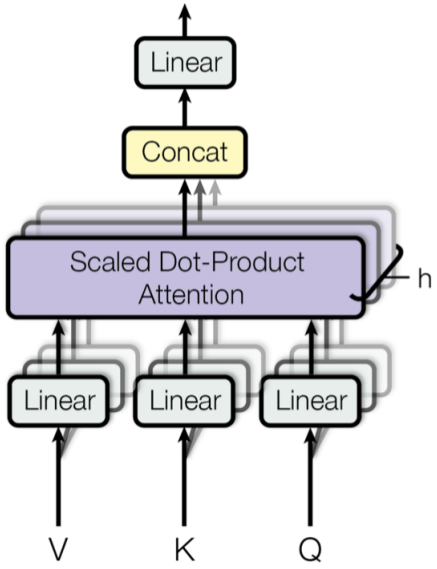


3.12 Irudia: Scaled dot product attention. [Vaswani et al., 2017]

Atentzio hau *buru* desberdinez osatuta dago, hau da, prozesu berdina egingo du matrize berdinekin baina pisu desberdinekin, sekuentziaren elementuen arteko erlazio eta menpekotasun desberdinak identifikatuz. Honi Multiheaded Attention, Self Attention edo euskaraz buru anizkoitzeko atentzioa deitu ahal zaio.

Atentzio honetan, Q eta K matrizeen arteko biderketa egiten da eta honen emaitza sekuentzien luzeraren erro karratuaz zatitzen da, pisuak lortuz. Zatidura hori egiten da matrizeen biderketa handiegiak ez egiteko. Ondoren, *softmax* funtzioa erabiltzen da pisuak normali-

zatzeko eta V matrizearekin biderkatzen dira. 3.12 irudian azalpen honen errepresentazioa ikus daiteke. Eragiketa hauek buru desberdinetan egiten direnez, bakoitzak irteera bezala ematen duen matrize guztiak konkatenatzen dira. Amaitzeko, parametroz osatutako matrize batekin biderkatzen da hasierako dimentsioetara bueltatzeko (3.9 ekuazioa). Horrela, buru bakoitzetik lortutako menpekotasunak bateratzen dira ereduak erabiltzen dituen embeddingetara. 3.13 irudian geruza honen errepresentazioa ikus daiteke.



3.13 Irudia: Multiheaded attention geruza. [Vaswani et al., 2017]

Ondoren, sare neuronal sakonen abantailak aprobetxatuz, elementuz elementuko FFNN bat erabiltzen da, atentziotik lortutako embeddingetan loturak aurkitzeko.

Self-attention eta FFNN pausoak eta gero Add and Norm deritzon geruza aurkitzen da (ikus 3.10 irudia). Geruza honi konexio erresiduala ere deritzo. Geruzako embedding berria lortu eta gero, geruzatik sartu baino lehen zuen embeddinga gehitzen zaio eta normalizatzen da (3.10 ekuazioa, non X jatorrizko sarrera den eta Y aurreko geruzaren irteera den), horrela balioak ez dira gehiegi aldatzen eta hein baten barruan mantentzen dira, ikasketa prozesua azkartuz eta orokortzeko gaitasuna hobetuz. Gainera, hasierako positional encoding pausoan sartutako informazioa mantentzeari laguntzen dio.

$$Add\ and\ Norm = norm(X + Y) \tag{3.10}$$

Deskodetzailearen geruza bakoitzak elementu berdinak dauzka, baina *Encoder-Decoder attention* deritzon geruza gehitzen zaio. Kodetzaileko azken geruzako self-attentionaren K

eta V matrizeak erabiltzen dira deskodetzaileko atentzio honetan. Horrela, jatorrizko hizkuntzaren erlazioen informazioa deskodetzerakoan erabiltzen da iragarpenak hobetzeko, beste atentzio geruzarekin batera, helburu hizkuntzaren menpekotasunak identifikatuko dituenak. Kodetzailean bezala, elementuz elementuko FFNNa amaieran egongo da eta *add and norm* geruza bakoitzaren amaieran egongo da.

Deskodetzaileari pasatzen zaizkion sekuentziak *positional encoding* ere aplikatzen zaie, baina i. hitza iragartzerakoan hitz horretatik aurrerako hitz guztiak maskaratzen dira eta horrela ereduak ezin izango du etorkizuneko hitzak erabili informazio gehigarri bezala, bakarrik aurrekoak.

Deskodetzailearen irteerari transformazio lineal bat aplikatzen zaio eta irteera bezala *logits* deritzon bektorea du. Deskodetzaileak entrenamenduko hitz posible guztien artean bat aukeratzeko logits bektorea erabiltzen da. Hitz posible horiek modeloak entrenamenduko datuetan eskuragai dituen hitz desberdinak dira, hiztegi deritzona. Ereduaren irteera hiztegiak iragarri ditzakeen hitzetaz osatuta dago. Hitz bakoitzari zenbaki bat dagokio. Logits bektoreari softmax funtzioa aplikatu eta gero, hitz bakoitzaren irteera izateko probabilitateak lortzen dira eta probabilitate handiena duen indizeari dagokion hitza ereduaren irteera izango da.

3.3 Hitzen errepresentazioa

Hitzak ordenagailu batean stringetan adierazi ohi dira, baina hauekin ezin dira eragiketa matematikoak egin. Sare neuronaletan zenbakizko bektoreak erabiltzen dira ataza desberdinak burutzeko. Zailtasunak hitzak bektorizatzean agertzen dira. Jarraian teknika desberdinak aipatzen dira.

Hasierako hurbilpena *one-hot encoding* bitartez lortzen diren bektoreak dira. Hitzen hiztegi bat emanda, hiztegiaren tamaina adina luzerako bektoreak sortzen dira. Hitz bakoitzari indizearen posizioan 1 jartzen zaio 0-ko bektoreetan (3.14 irudia). Teknika honek daukan eragozpena da ez dela eraginkorra. Oso handia den hiztegi batean, demagun 10.000 hitzetakoa dela, hitz bakoitza errepresentatzeko 10.000 luzerako bektoreak erabili behar dira eta horietako bat izan ezik, gainerakoek 0-ak dira. Gainera, zenbaki horiek ez dute hitzen arteko erlazioak adierazten.

Hurrengo hurbilpena hitz bakoitzari zenbaki bat baino ez esleitzea da. Aurreko kasuan bezala, hiztegi bateko hitz bakoitzari zenbaki desberdinak jarriko zaizkie (3.15 irudia).

Etxea urdina da	etxea	1	0	0
	urdina	0	1	0
	da	0	0	1

3.14 Irudia: One hot encoding.

Teknika honen desabantailak agerikoak dira. Zenbaki bakunekin ezinezkoa da hitzen arteko erlazioak gordetzea. Gainera, sare neuronal lineal batean hauek erabiliz gero, hitzen antzekotasuna eta haien errepresentazioen arteko erlaziorik existitzen ez denez, sarearen ezaugarri bakar horren parametroak ez dute inolako esanahirik eskaintzen.

Etxea urdina da	etxea	1
	urdina	2
	da	3

3.15 Irudia: Zenbaki bakarrez egindako kodeketa.

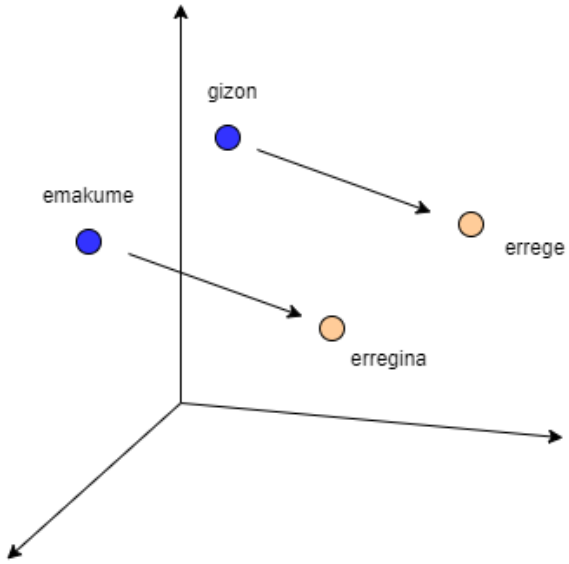
Azkenik, gaur egun emaitza onak ematen dituen eta erabiltzen den teknika *word embeddingak* dira. Teknika honen bitartez, parametro batekin adierazitako luzeraz zenbaki errealez osatutako embeddingak erabiltzen dira hitzak errepresentatzeko (3.16 irudia). Embedding hauen balioak eskuz jarri beharrean, aurreko teknkiketan egiten den moduan, eredu batean ikasten diren parametroak dira. Embedding hauen luzera 8-tik 1024-ra arte doaz normalean, baina luzera gero eta handiagoa izan, orduan eta erlazio zehatzagoak lortu daitezke. Aldi berean, embedding luzeagoak erabiliz gero, datu gehiago erabili behar dira bektoreak entrenatzeko. Bektorearen posizio bakoitza ezaugarri bati dagokio, baina ezaugarri hauek abstraktuak dira eta dimentsioen arteko erlazioak lortzeko erabiltzen dira.

Erlazio hauen adibidea gizon-emakume hitzen arteko analogiarekin ikus daiteke. Hiru dimentsiotara ekarritako embeddingak erabiliz (ereduetan erabilitako embeddingen dimentsioak askoz handiagoak dira), hurrengo erlazioak sortzen dira errepresentazioen artean, 3.17 irudian agertzen direnak. Kasu sinple honetan, bektoreen batura eta kenketaren bitartez analogia egokiak lor ditzakegu, 3.11 formulan agertzen den moduan. Ikusten den moduan, analogia honetan bi erlazio agertzen dira: gizon-emakume eta errege-erreginen eremu semantikoko hitzen artean.

Etxea urdina da	etxea	0.9741	-0.1427	0.0771
	urdina	-0.1581	0.4961	0.2637
	da	0.0854	0.8414	-0.4716

3.16 Irudia: Zenbaki bakarrez egindako kodeketa.

$$\overrightarrow{erregin} + (\overrightarrow{gizon} - \overrightarrow{emakume}) \approx \overrightarrow{errege} \tag{3.11}$$



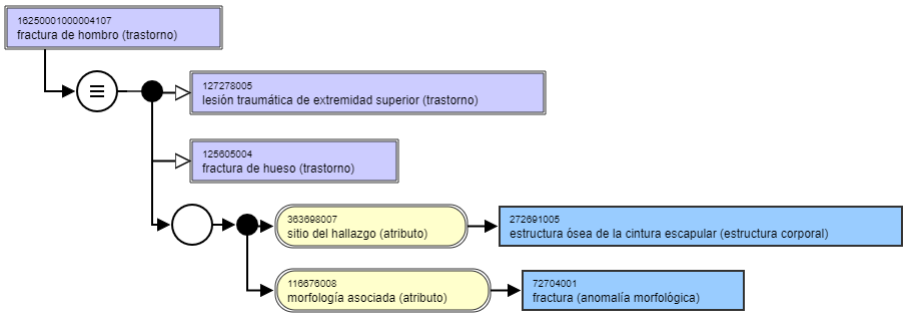
3.17 Irudia: Gizon-emakume analogiaren adibidea.

Embeddingak entrenatzeko maiz erabiltzen diren teknikak *word2vec* [Mikolov et al., 2013], GloVe [Pennington et al., 2014] eta *fastText* [Joulin et al., 2016] dira. Lan honetan, edonola ere, 3.4 atalean azaltzen den ezagutza basearen edo ontologiaren hitzen embedding entrenatuak erabiltzen dira ereduan erabiltzeko, gero azalduko den moduan. Hau egiteko aurreko tekniken word2vec delakoa erabiltzen da, baina ontologietan aplikatuta (4.3 atala).

3.4 Ezagutza baseak

Ezagutza baseak informazioa modu ordenatuan gordetzen dituzten datu baseak dira. Hasieran, ezagutzan oinarritutako sistemetan erabiltzen ziren domeinu bateko ataza bat ebazteko ezagutzari zentzua emateko eta inferentzia motorrak informazio hori erabili ahal izateko. Gaur egun, modu hierarkikoan ordenatzen diren datu multzoak erabiltzen dira aplikazioetan, ezagutza baseetako elementuen erlazioak, hain zuzen ere. Hizkuntzaren prozesamenduaren arloan oso ezaguna den bat WordNet da [Miller, 1998], hizkuntza desberdinetarako hitzen arteko erlazioak gordetzen dituena.

Lan honetan domeinu biomedikoko termino desberdinak batzen dituen SNOMED CT ontologia erabiltzen da. WordNet bezala, ontologia honek gordetzen dituen terminoen eremu semantikoen arteko erlazioak existitzen dira. WordNet-en erlazioak orokorrak dira (sinonimoak, antonimoak, hiperonimoak, meronimoak...). SNOMED CT-n, aldera, erlazio hauek testuinguru biomedikoan agertzen direnak dira (“da” erlazio taxonomikoa, aurkikuntza lekua, eragilea...). Hona hemen SNOMED CT ontologiaren sarearen adibide bat (3.18 irudia).

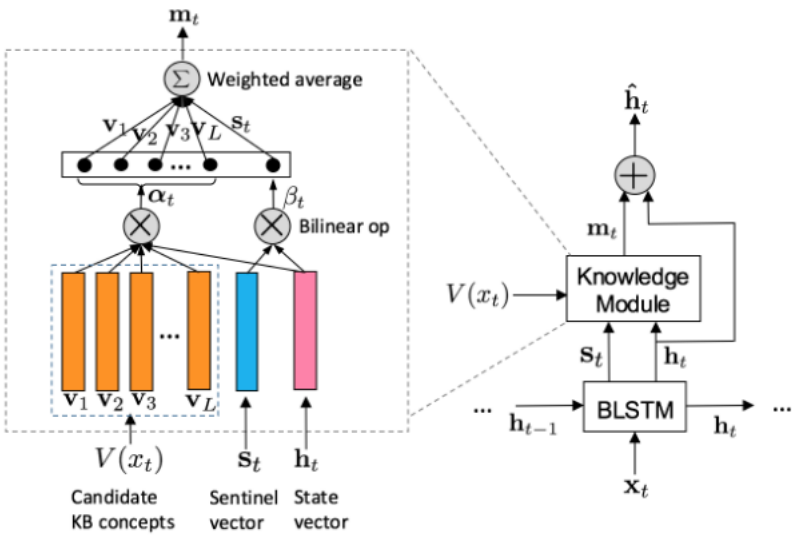


3.18 Irudia: SNOMED CT ontologiaren terminoen arteko erlazioen adibidea.

3.4.1 Kanpo ezagutza sartzeko teknikak

Transformer arkitektura berria izanik, oraindik ez dira ezagutza baseak txertatzeko lanak argitaratu; bai ordea, LSTM arkitekturarako [Yang and Mitchell, 2019].

LSTM arkitekturan, 3.2.1 atalean azaldu den moduan, iterazio bakoitzean egoera ezkutu bat eramaten da iterazio bakoitzean, beste bektore batzuekin batera ate funtzioa egiten duena.



3.19 Irudia: KBLSTM arkitektura. [Yang and Mitchell, 2019]

Kanpo ezagutza sartzeko LSTMko bertsio honetan [Yang and Mitchell, 2019], KBLSTM deitzen diotena (3.19 irudia), egoera ezkutua (h_t) erabiltzen dute, LSTMaren informazio garrantzitsuena daramana. Sentinel vector (s_t) deritzoten memoria atea, egoera ezkutua eta sarrerarekin kalkulatzen da. Bi hauekin testuinguru bektore bat lortzen da. Bestalde, V_{x_t} bektoreak momentuko hitzaren embedding hautagaiak dira. Hauek egoera ezkutuarekin konbinatuz atentzio bektore bat lortzen da. Honekin batezbeste ponderatuaren bitartez bektore bat lortzen da. Egoera ezkutuaren informazioa gehiegi ez galtzeko, Knowledge Modulutik lortutako bektorea egoera ezkutuarekin gehitzen da eta hori egoera berria bihurtzen da.

4. KAPITULUA

Metodologia

Atal honetan proiektuaren diseinua eta inplementazioaren nondik norakoak azaltzen dira. Datasetaren aukeraketa eta aurreprozesaketa deskribatzen dira, ereduan erabili ahal izateko. Lan honetan testu biomedikoekin lan egiten denez, domeinu horietako corpusak aukeratu dira entrenamendurako eta ebaluaziorako.

4.1 Datasetaren aukeraketa

Itzultzaile automatikoaren eredua entrenatzeko, erabili behar den datu kantitatea oso handia izaten da. Hori dela eta, corpus handiak erabili behar dira. Corpus bat testu asko dauzkan biltegia da. Eredua ikasketa gainbegiratuaren bitartez entrenatuko denez, bi hizkuntzatan dauden corpus paraleloak aukeratu behar dira. Gainera, esaldiak lerrokatuta egon behar dira. Ereduak sekuentziaka (edo esaldika) ikasten duenez, testu handietan esaldi bakoitzaren itzulpen zehatza lerrokatuta izatea garrantzitsua da. Corpus paralelo bezala zabaltzen diren corpus gehienak dagoeneko lerrokatuta datoz. Hauek eskuz edo automatikoki eraiki dira. Eskuz lerrokatuta daudenak gehienetan zuzeneko itzulpen bidez sortutako corpusak dira eta guztiz fidagarriak dira, adituek itzuli edo lerrokatu dituztelako. Automatikoki lerrokatutako testuetan programa edo eredu batek bi esaldien arteko baliokidetasuna neurtzen du [Simard and Plamondon, 1998]. Hori dela eta, esaldi bakoitzari zenbaki bat esleitzen zaio eta fitxategi batean hizkuntza bateko esaldi baten eta beste hizkuntzako esaldien arteko lerrokatze maila agertzen da. Bi lerrokatze mota daude: guztiz lerrokatuta daudenak eta esaldien artean gainjartzea dutenak. Esaldi baten itzulpenera-

ko bi esaldi edo gehiago erabili izan badira, baliteke itzulpen zuzena izatea baina hitz gehiago erabilita edo esaldien arteko gainjartzea egotea. Gainjartze honek adierazten du esaldien artean informazioaren errepikapena dagoela, hau da, informazio berdina esateko bi modu desberdinetan adierazi da. Amaitzeko, fitxategian bi esaldien arteko lerrokatzea ez dagoela ere adierazten da. Bi esaldien arteko itzulpena osoa ez denean eta informazio galera dagoenean, programak adierazten du ezin dela kontsideratu esaldi hori bestearen itzulpen onargarria dela.

Proiektu honetan biomedikuntzaren domeinuko esaldiak eta terminologia itzultzeko ahalmena aztertzen denez, entrenatzeko eta probatzeko corpusen domeinua biomedikoa izan behar da. Hauek lortzeko, *Association for Computational Linguistics* (ACL) elkarteak 2020an egindako *FIFTH CONFERENCE ON MACHINE TRANSLATION* (WMT20) konferentziako workshop batean [1], domeinu biomedikoan itzulpen automatikoa burutzeko atazarako eskuragarri dauden corpusak erabiltzen dira. Corpus horien barruan, testu batzuk entrenatzeko ematen dira eta beste batzuk ebaluatzeko. Bien arteko desberdintasuna esaldi kopurua da. Itzulpen automatikoko ereduak entrenatzeko ahalik eta esaldi gehienen erabiltzen dira; ebaluatzeko hiru mila esaldi inguru baino ez dira behar emaitza esanguratsuak izateko.

Entrenatzeko, domeinu biomedikoko corpustzat Medline aukeratu da. Medline informazio biomedikorako erreferentziak biltzen dituen datu base bibliografikoa da, Estatu Batuetako Medikuntzako Liburutegi Nazionalak kudeatua [2]. Bertan, artikulu akademikoak eta biomedikuntzaren arloko beste dokumentuak gordetzen dira. 4.1 taulan entrenatzeko Medline corpusaren adibide batzuk ikus daitezke.

Domeinuko esaldi gutxi daudenez, ereduan emaitzak hobetzeko asmoz, domeinutik kanpoko esaldiak ere erabiltzen dira eredua entrenatzeko, 4.3 taulan ikusten den moduan. Erabilera orokorreko corpus handiak erabili daitezke, baina tamainagatik eta aurreprozesatzeko erraztasunagatik Europarl corpusa aukeratu dugu [Koehn, 2005]. Corpus honek Europako Parlamentuan parlamentukide desberdinek esandako esaldi transkribatuak dauzka, alor desberdinei buruz doazenak. 4.2 taulan Europarl corpuseko adibide batzuk ikus daitezke.

Eredua ebaluatzeko Medlineko corpusa ere erabiliko da, baina beste esaldi batzuk erabilita. Gainera, corpus honek daukan abantaila bat alor biomedikoko terminoak batzen dituen corpus paralelo bat daukala. Horrela, ereduak terminoak itzultzeko ahalmena esaldiez aparte ebaluatu daiteke.

[1] http://www.statmt.org/wmt20/
[2] https://www.nlm.nih.gov/bsd/medline.html

	Gaztelera	**Ingelesa**
1	Pancreatitis aguda farmacológica.	Drug-induced acute pancreatitis.
2	Hemorragia digestiva alta por ingestión de un comprimido en blíster.	Gastric hemorrhage after ingestion of a blister-wrapped tablet.
3	Masa parotídea como forma de presentación de un linfoma de Hodgkin.	Parotid mass as presenting symptom of Hodgkin lymphoma.
4	Lactante con tortícolis adquirida: cavernoma gigante cerebeloso.	Post-natally acquired torticollis: Giant cavernous cerebellum.
5	Análisis coste-utilidad de apixabán frente al ácido acetilsalicílico en la prevención del ictus en pacientes con fibrilación auricular no valvular en España.	Cost-effectiveness analysis of apixaban versus acetylsalicylic acid in the prevention of stroke in patients with non-valvular atrial fibrillation in Spain.

4.1 Taula: Medline corpuseko entrenamendurako adibide batzuk.

4.2 Aurreprozesaketa

Datasetaren aukeraketaren atalean azaltzen denez, corpus guztiak ez datoz erabiltzeko prest. Gainera, normalizatzeko bateratasun falta dela eta, corpus bakoitza modu desberdinean aurreprozesatzeko beharra dago.

Entrenatzeko Medline corpusa dagoeneko lerrokatuta zetorren, beraz, egin behar izan den aurreprozesaketa bakarra fitxategi bakar batetik bi fitxategi lerrokatuetara banatzea izan da, non fitxategi baten lerro baten esaldiaren itzulpena, beste fitxategiko lerro berdinean aurkituko den. Terminologiako corpusean csv fitxategi batetik bi fitxategi lerrokatuetara pasatu behar izan da. Entrenatzeko Europarl corpusarekin dagoeneko prozesu hau eginda zegoen.

```
30777411        8 <=> 9 OK
30777411        9 <=> 10        OK
30777411        10 <=> 11       OK
30777411        11,12 <=> 12    OK
30777411        13 <=> 13       OK
30777411        14 <=> 14       OK
30777411        15 <=> 15       OK
30777411        16 <=> 16       OK
30741657        1 <=> 1,2       NO_ALIGNMENT
30741657        2 <=> 3,4       OVERLAP
```

4.1 Irudia: Lerrokatze automatikoak sortutako fitxategiaren zatia.

	Gaztelera	**Ingelesa**
1	Reanudación del período de sesiones	Resumption of the session
2	Declaro reanudado el período de sesiones del Parlamento Europeo, interrumpido el viernes 17 de diciembre pasado, y reitero a Sus Señorías mi deseo de que hayan tenido unas buenas vacaciones.	I declare resumed the session of the European Parliament adjourned on Friday 17 December 1999, and I would like once again to wish you a happy new year in the hope that you enjoyed a pleasant festive period.
3	Como todos han podido comprobar, el gran "efecto del año 2000"no se ha producido. En cambio, los ciudadanos de varios de nuestros países han sido víctimas de catástrofes naturales verdaderamente terribles.	Although, as you will have seen, the dreaded 'millennium bug' failed to materialise, still the people in a number of countries suffered a series of natural disasters that truly were dreadful.
4	Sus Señorías han solicitado un debate sobre el tema para los próximos días, en el curso de este período de sesiones.	You have requested a debate on this subject in the course of the next few days, during this part-session.
5	A la espera de que se produzca, de acuerdo con muchos colegas que me lo han pedido, pido que hagamos un minuto de silencio en memoria de todas las víctimas de las tormentas, en los distintos países de la Unión Europea afectados.	In the meantime, I should like to observe a minute' s silence, as a number of Members have requested, on behalf of all the victims concerned, particularly those of the terrible storms, in the various countries of the European Union. Please rise, then, for this minute' s silence.

4.2 Taula: Europarl corpuseko entrenamendurako adibide batzuk.

Medlineko ebaluaziorako esaldien corpusarekin zenbait fitxategi etortzen ziren. Lehenengo fitxategiak dokumentu bakoitzaren izena identifikadore batekin erlazionatzen zuen. Hurrengo fitxategiak gazteleraz eta ingelesez zetozen esaldiak ziren. Lerro bakoitzean dokumentuaren izena, dokumentu horretako esaldiaren zenbakia eta esaldi bera, elementu bakoitza tabulazioz banatuta. Azkenik, lerrokatze dokumentu bat zetorren, non dokumentu bateko identifikadore bat eta esaldi batzuen arteko lerrokatze maila adierazten zen (4.1 irudia). Aurreprozesaketa honetan, OK lerrokatze maila zuten esaldiak baino ez ziren aukeratu ebaluazio-multzoa sortzeko.

Behin corpus guztiak erabiltzeko prest daudela, testuan dauden SNOMED CTko terminoak erauzteko, FreeLing-Med programa erabili da [Oronoz et al., 2013]. Programa honek

Dataseta	Esaldi kopurua	Token bakarrak	
Europarl	1.965.734	es 422.630	en 308.978
Medline	285.082	es 213.877	en 175.168

4.3 Taula: Entrenatzeko dataseten ezaugarriak.

esaldiak prozesatzen ditu eta SNOMED CT ontologiaren termino bat aurkitzen badu, bere identifikadorea esleituko zaio. Termino hauek hitz batekoa edo hitz batzuetakoa izan daiteke. FreeLing-Medek KAF formatuko fitxategi batean bueltatzen du informazio guztia. KAF formatua XML oinarri duen formatua da [Bosma et al., 2009].

```
<term tid="t6973" lemma="right_ventricular_hypertrophy" pos="NA">
  <span>
    <target id="w8610"/>
  </span>
  <externalReferences>
    <externalRef resource="SCT_es_INT_20130731" reference="89792004" reftype="disorder">
      <externalRef resource="UMLS-2010AB!" reference="C0162770"/>
    </externalRef>
  </externalReferences>
</term>
```

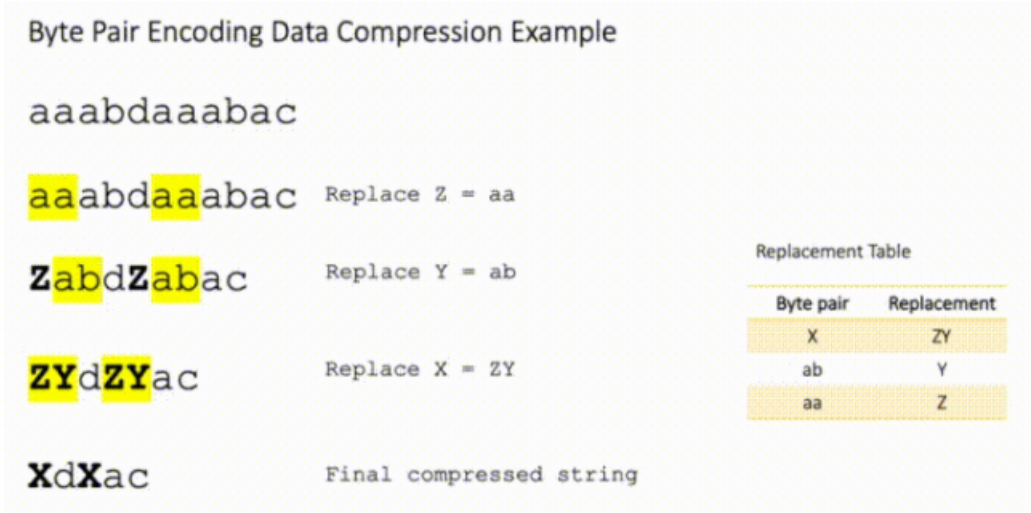
4.2 Irudia: KAF fitxategiko termino baten erauzketaren adibidea.

FreeLing-Med programak burutzen zituen atazen artean esaldien identifikazioa, hitzen tokenizazioa eta esaldien hitzen analisi sintaktikoa (objektu zuzena, subjektua, aditza, aditzondoa...). Lan honen atazarako, ostea, programa honen atazarik garrantzitsuena pasatzen zaizkion corpusetan SNOMED CT ontologiako terminoen identifikazioa da (4.2 irudia). Hauek erabilgarriak izango dira ondoren 4.4 atalean sartutako hobekuntzan erabiliko baitira.

4.2.1 BPE

BPE kodeketa 1994. urtean datuak konprimitzeko teknika proposatu zen [Gage, 1994]. Honen funtzionamendua iteratiboki maizen agertzen diren ondoz ondoko byte bikoteak byte berri batean ordezkatzean datza. 4.3 irudian prozesu honen adibidea ikus daiteke.

Itzulpen automatikorako atazan lehenengo aldiz 2015. urtean proposatu zen [Sennrich et al., 2015]. Eredu baterako eskuragarri dauden hitzak entrenamendurako corpusean agertzen direnak dira. Probatzerako momentuan gerta daiteke entrenamenduan agertu ez



4.3 Irudia: Datu konpresiorako BPE kodeketa.

den hitz bat agertzea eta eredua ez jakitea zein hitz erabili. Horretarako <unk> tokena erabiltzen zen. Hau ekiditeko BPE kodeketa corpusa aurreprozesatzeko erabili zen, ataza honetara egokituta. Hauek dira itzulpen automatikorako BPE algoritmoak jarraitzen dituen pausoak:

1. Hiztegia hasieratu.
2. Corpuseko hitz bakoitza karaktereetan banatu, hitz mugetan <w/> token berezia gehituz.
3. Iteratiboki hiztegiko token guztietan karaktere pare guztiak kontatu.
4. Gehien agertu den bikote parea token guztietan batu eta hau hiztegira gehitu.
5. Laugarren pausoa errepikatu iterazio kopuru maximora edo hiztegiaren tamaina maximora heldu arte (hiperparametro batean adierazita).

BPE bitartez tokenizatutako hitz baten adibidea hurrengoa da: “liburutegiko” hitza BPE tokenizatuz gero hurrengo tokenak geratuko lirateke: *_liburu* + *tegi* + *ko*. Azpibarrak adierazten du token horrekin hitz berria hasten dela. Horrela, token hauek deskodetzeko prozesua tribiala bihurtzen da: zuriuneak kendu eta azpibarrak zuriuneekin ordezkatu.

Prozesu honen bitartez, bi hizkuntzen arteko hitzak itzultzeko ez da beharrezkoa izango <unk> tokena erabiltzea, ereduak ez badu hitza ezagutzen.

Lan honetan, BPE kodetzeko eredua entrenatzeko eta hiztegia sortzeko bi hizkuntzetan dauden corpusak batu dira, bi hizkuntzek token berdinak erabili ahal izateko. Honek ereduaren ikasketa prozesua errazagoa egiten du.

Corpusetan BPE kodeketa aplikatzerakoan SNOMED CT hitzei esleitutako identifikadoreak ere banandu egin behar dira. Egoera hau konpontzeko, hitz baten identifikadorea kodeketa eta gero sortutako tokenei esleituko zaie. Horrela, ereduari adierazi egingo zaio token hori termino biomediko baten parte dela.

Demagun aurreko adibideko "liburutegiko"|ID052 hitza daukagula eta ID052 identifikadorea duela. BPE kodeketa egin eta gero *_liburu* + *tegi* + *ko* tokenak geratzen zaizkigu eta bere jatorrizko hitzaren identifikadore berdina esleituko zaie; hortaz, *_liburu*|ID052 + *tegi*|ID052 + *ko*|ID052 geratuko litzateke.

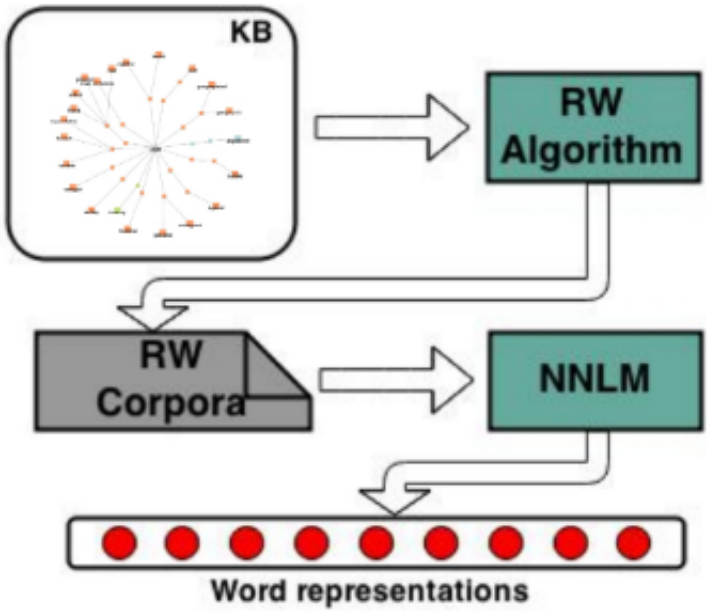
4.3 Ezagutza baseen terminoen embeddingak entrenatuz

SNOMED CT ontologiako terminoen informazioa 4.4 ataleko ereduan sartzeko, SNOMED CT terminoen errepresentazio jarraituak edo *word embeddingak* lortu behar dira. Aurrekarietako 3.3 atalaren amaieran hitz errepresentazio hauek entrenatzeko metodo desberdinak aipatu egin dira. Lan honetan, ordea, ezagutza baseekin lan egiten denez, aparteko teknika erabili da.

Teknika honek ezagutza base bat erabiltzen du sasicorpus bat sortzeko eta hortik *word2vec* erabili daiteke termino bakoitzaren errepresentazio jarraitua ikasteko [Goikoetxea et al., 2015]. Prozesu osoaren diagrama 4.4 irudian ikus daiteke.

Sasicorpusa sortzeko, *random walks* edo ausazko bideak lortzen dira. SNOMED CT ontologian aplikatuta, algoritmo honen arabera, hasierako termino bat ausaz aukeratzen da hasiera puntu bezala. Ikusi den bezala, termino bakoitzak erlazio desberdinak ditu beste terminoekin. Hauek bide bezala erabiliko dira eta algoritmoak bide horietako bat ausaz aukeratuko du eta erlazio hori eramaten duen terminora joango da. Prozesua errepikatuko da termino kopuru jakin batera heldu arte (hiperparametroz adierazita). Horrela, terminoen errepresentazioak lortzeko sasicorpusak lortu egin dira.

Ondoren, sasicorpusa *word2vec* motako sare neuronalean entrenatzeko erabiliko da eta hortik termino bakoitzaren embeddingak lortzen dira.



4.4 Irudia: Ezagutza baseen terminoen embeddingak ikasteko prozesua. [Goikoetxea et al., 2015]

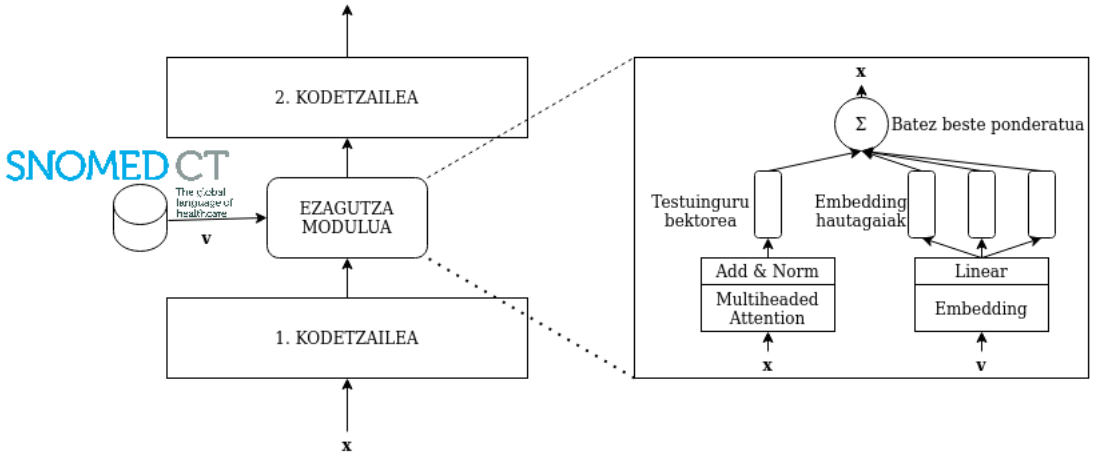
4.4 Ereduaren diseinua

Ereduaren diseinua 3.4.1 atalean azaldutako teknikan eta Transformerreko arkitekturan eta teknologietan oinarritzen da.

Esperimentu honetan, Transformer arkitekturari Ezagutza Modulua edo Knowledge Module deritzon blokea sartzen zaio kodetzailearen lehenengo eta bigarren geruzen artean (4.5 irudia). Knowledge Module honetan bi atal desberdin bereiz daitezke: barruko egoera eta kanpoko informazioa.

Barruko egoera errepresentatzeko Multiheaded Attention-a erabiliko da. Kanpo ezagutza sartzeko tekniken 3.4.1 atalean, Sentinel Vector deritzon bektorea lortzeko LSTMaren egoera ezkutua erabiltzen zen. Hori dela eta, Multiheaded Attention-a erabiltzea erabaki da, hau hitz bakoitzaren testuinguru bektorea lortzen baitu sekuentziaren beste hitzekiko. Geruza hau eta gero, Transformer arkitekturan bezala, Add and Norm geruza jarri da. Lortzen den bektorea testuinguru bektorea deritzo eta sekuentzia baten hitz bakoitzaren testuinguru embeddinga gordetzen du.

Bestalde, kanpoko egoera adierazteko SNOMED CT ontologiatik datozen terminoen informazioa sartzeko 4.2 atalean azaldu den aurreprozesaketa aplikatu zaion fitxategia erabiltzen da. Honek, jatorrizko hizkuntzako corpusaren tokenen SNOMED CTko identifika-



4.5 Irudia: Proposatutako hobekuntza.

doreak dauzka, 4.2.1 atalean egindako moldaketa jarraituz. Hasieran, sekuentziari dagokion token bakoitzaren identifikadoreak geruzaren sarrera izango dira. Hauek Embedding geruza batetik pasatuko dira, identifikadoreetatik 4.3 atalean entrenatutako SNOMED CT terminoen embeddingetara bihurtzeko. Embedding hauek ereduan erabiltzen den dimentsioetara egokitzeko geruza lineala erabiltzen da. Termino bakoitzeko, gehienez, hiru embedding desberdin egon daitezke, ontologian dauzkan interpretazioen araberakoa. Hiru embedding hautagai baino gutxiago baldin badaude, identifikadorea faltatzen den bektoreetan beste identifikadore bat erabiltzen da ez dela inolako informaziorik sartzen adierazteko eta horrela informaziorik ematen ez duen embeddinga sartuz.

Barruko eta kanpoko informazioa batzeko, 3.4.1 atalean azaltzen diren eragiketa berdinak erabiltzen dira sekuentziaren token bakoitzerako: embedding hautagai bakoitzetik eta testuinguru bektorearen artean atentzio zenbaki bat kalkulatzen da eta hori bektore hautagaiaren pisua bihurtzen da. Testuinguru bektoreak ere bere pisua du. Pisu hauekin, batez besteko ponderatua egiten da eta behin betiko bektorea lortzen da.

4.5 Inplementazioa

Esperimentuaren diseinuko inplementaziorako Python lengoaiaren 3.6 bertsioa erabili da. Eredua inplementatzeko hurrengo liburutegiak erabili dira:

- **torch**, sare neuronalen inplementaziorako.
- **math**, tentsoreen arteko eragiketak egiteko.

- **time**, ereduen epoken luzera neurtzeko.
- **copy**, objektuen kopiak egiteko.
- **dill**, pickle liburutegia bezala, ereduaren parametroak gordetzeko eta probetarako erabiltzeko edo horietatik entrenamendua jarraitzeko.
- **easydict**, hiztegien bertsio erabilgarrigagoa da, ereduaren hiperparametroak kudeatzeko.
- **pandas** eta **torchtext** , eredura pasatzen zaizkion fitxategiak eredura moldatzeko.
- **os**, fitxategiak sortzeko, idazteko, irakurtzeko eta ezabatzeko.
- **nltk**, BLEU metrika kalkulatzeko.
- **sentencepiece**, corpusaren gainean BPE kodeketa burutzeko.
- **ElementTree**, XML fitxategiak parseatzeko.

Inplementazioaren abiapuntua Transformer bat garatzeko tutorial bat izan zen. Bertan, oinarrizko aurreprozesaketa burutzen zen eta arkitekturaren modulu desberdinak aurkitzen ziren. Hala ere, errore batzuk zeuzkan eta konpondu behar izan ziren. Goitik behera kodeak lerro bakoitzean egiten zuena aztertu behar izan zen eta soberan zegoena kendu zen. Inplementazio honek paketeak edo *mini-batchak* sortzeko optimizazio bat erabiltzen zuen.

Batchak sarrerako datuen instantzia paketeak dira. Lan honetan, milioi bat esaldi baino gehiago daude eredua entrenatzeko epoka, edo iterazio, bakoitzean. Entrenamendu fasea hobetzeko, instantzia kopuru bat (hiperparametroz adierazita) paketeetan sartzen da eta epoka bakoitzeko parametroen eguneraketa bakarra egin beharrean, sortutako mini-batch bakoitzarekin eguneraketa egiten da epoka bakar batean.

Batchak luzera berdintsuko esaldiekin batzen dira, matrizeak ahalik eta gehienen betetzeko eta *padding* aldaketa handirik ez egoteko. Ondoren, iteratzaile bat sortzen da entrenamendurako, baina orden berdinean sartu beharrean, ausaz aukeratzen dira paketeak. Horrela, ereduak ez du esaldien arteko patroirik ikasten.

Padding prozesua sekuentzietaz osatutako sarrerako matrizeak betetzean datza. Batchak dimentsio jakin batekoak dira eta ikasketa osoan zehar finko mantentzen dira. Sarrerako esaldiak, ordea, luzera desberdinekoak dira. Beraz, esaldiaren luzera matrizearena baino

txikiagoa denean, <pad> tokena gehitzen zaio, eta ereduaren ikasketan zehar hau maskaratzen da, ez baitu informaziorik eskaintzen.

Ereduan hobekuntza sartu ahala, inplementazioa ere aldatu behar izan zen. Esaterako, kontuan hartu behar izan zen fitxategi estra bat sartzen zela, token bakoitzaren SNOMED CT identifikadoreak dituena, hain zuzen ere. Gainera, modulu berri bat sartu zen kodetzailearen zatian kanpoko ezagutza sartzeko.

Ereduaren ebaluaketa egiteko hasieran Google Colab plataforman burutu zen. Proba hauek esaldi laburrekin burutu ziren eredua ikasten zuela bermatzeko. Behin hori lortzen zela ikusita, IXA zerbitzarietan probatzera pasatu zen eta 4.1 atalean adierazitako esaldiekin probak burutu ziren.

CUDA teknologiari esker (Compute Unified Device Architecture) bai Google Colab-eko bai IXAko grafikak erabiltzea erraza izan zen. *torch* liburutegiak CUDA erraz erabiltzeko funtzioak dauzka.

Emaitzen 5.2 atalean ematen diren BLEU metrikak lortzeko, hurrengo hiperparametroak erabili ziren ereduan:

- Epoka kopurua: 10.
- Ereduko tokenen embedding luzera: 512.
- Ezagutza baseko terminoen embedding luzera: 300.
- Kodetzaileen eta deskodetzailearen geruza kopurua: 6.
- Multiheaded Attentioneko geruza kopurua: 8.
- Dropout probabilitatea: 0,1.
- Batch tamaina: 256.
- Learning rate-a: 0,0001.

Corpusen aurreprozesaketa egiteko *scriptak* sortu dira. Egindako ataza nagusiak karaktere arraroen garbiketa, XML eta KAF fitxategien parseatzea eta BPE kodeketa izan dira.

5. KAPITULUA

Emaitzak

Atal honetan egindako esperimentuak deskribatuko dira eta hauen emaitzak emango dira.

Egindako esperimentuak 4.1 ataleko datasetekin burutu egin dira. Corpus hauek elebidunak izanik gazteleratik ingelesera eta alderantziz egin dira probak ereduak bi hizkuntzak ikasteko zailtasunak ikusi ahal izateko. Aurretik azaldu bezala, bi dataset desberdin erabili dira entrenamendurako: Medline eta Europarl. Ereduaren ahalmena probatzeko Medlineko bi corpus desberdin erabili dira: bat terminoak besterik ez biltzen duena eta bestea testu biomediokoetan agertu daitezkeen esaldi osoak dituena.

5.1 BLEU

BLEU metrikak (bilingual evaluation understudy) itzulpen automatikoko ereduek itzulitako testuak ebaluatzeko balio du [Papineni et al., 2002]. Metrika hau 0 eta 1 bitarteko zenbakia da eta automatikoki itzulitako testuaren eta kalitate handiko itzulpenaren arteko berdintasuna adierazten du. 0 balioak adierazten du automatikoki itzulitako testuak ez duela hitz bat ere asmatu kalitate handikoarekin konparatuz. 1 balioak, ordea, itzulpen guztiz zuzena egin duela adierazten du.

Probatua izan da BLEU metrika eta pertsona batek itzulpen bati buruz duen iritzia bat egiten dutela. Esan beharra dago gizakiak egindako itzulpena ere ez dutela 1eko balioa.

5.2 Emaitzak

Ebaluazioaren emaitzak 5.1 taulan aurkitu daitezke.

		$es \rightarrow en$	$en \rightarrow es$
Baseline	Esaldiak	11,43	10,06
	Terminologia	9,17	9,82
Hobekuntza	Esaldiak	11,31	10,51
	Terminologia	9,23	10,27

5.1 Taula: Proben BLEU emaitzak.

Baseline-a edo oinarria hobekuntzarik gabeko Transformer arkitekturarekin [Vaswani et al., 2017] lortutako emaitzak dira. Hobekuntza, ordea, 4.4 atalean proposatutako hobekuntzarekin Transformer arkitekturak lortu dituen emaitzak dira. Bi ereduak 4.1 atalean deskribatutako corpusekin entrenatu dira. Horretaz gain, gaztelera eta ingelesaren arteko ikasketa aztertu da bi noranzkoetan.

Emaitzei begira, bi joera ikus daitezke, txikiak badira ere. Lehenengoa esaldien eta terminoen da. Gaztelaniatik ingelesera itzultzerakoan ereduak ingelesetik gaztelerara baino emaitza hobeak ematen ditu esaldiekin. Terminologiaren kasuan kontrakoa gertatzen da, hots, ingelesetik gaztelerara terminoak hobeto itzultzen ditu. Bigarrena emaitzen hobekuntzetan dago. Gazteleratik ingelesara itzultzerakoan, hobekuntzak ez dirudi aldaketa handiak ematen dituen. Esaldiekin emaitza zertxobait txarragoa da eta terminologiarekin pixka bat hobea da. Ingelesetik gaztelerazko itzulpenetan bi corpusekin hobekuntzak daude eta beste noranzkoan baino handiagoak dira. Honen zergatia ez da argia; baliteke SNOMED CT-k informazio gehiago eskaintzea ingelesez erabiltzen denean.

6. KAPITULUA

Ondorioak eta etorkizuneko lana

Atal honetan emaitzetatik atera daitezkeen ondorioak eta etorkizunean egingo diren hobekuntzak edota aldaketak azaltzen dira.

6.1 Ondorioak

Lan honetan zehar hasieran proposatuko helburu nagusiak bete dira. Itzulpen automatikoaren arloaren egoerako teknologiak (Transformer arkitektura, Multiheaded Attention-a, embeddingak...) ulertzea lortu da. Hortik, ezagutza baseen egitura aztertu dira eta Transformer arkitekturan sartzeko hobekuntza proposatu da. Diseinu honetatik *Pytorch* erabiliz Transformer arkitektura inplementatzea lortu da, hobekuntzarekin batera. Ezagutza basearri begira, SNOMED CT ontologia aztertu da eta termino bakoitzaren embeddingak lortu dira, ereduan erabiltzeko. Proposatutako hobekuntza ebaluatzeko, aztertutako domeinuarekin bat egiten duen Medline corpusa aurkitu eta aurreprozesatu da. Aurreprozesatze honetan, Freeling-Med programaren bitartez SNOMED CT-ko terminoak identifikatu dira corpusa erausiz. Gainera, domeinu orokorreko Europarl corpusa erabili da ereduaren entrenamendu fasea hobetzeko.

Lan honen emaitzak ikusita, ez dira espero zirenak lortu. Baseline-ko emaitzak baxu samarrak izan dira arloaren egoerarekin konparatuta. Honen arrazoia lanean zehar izandako denbora, datu kopuru eta memoria murriztapenengatik izan da. Hasieran, 20 epokarekin eredua entrenatzea proposatu zen, baina entrenamendu denbora luzeegia ez izateko (24 ordu baino gehiago proba baterako), epoka kopurua 10era murriztu zen. Horren ondorioz,

baliteke sistemak epoka nahikoa ez izana ikasketari aprobetxamendu guztia ateratzeko. Datu kopuruaren aldetik, domeinu biomedikoko entrenatzeko esaldi kopuruaren proportzioa %20a baino gutxiagokoa da.

Hala ere, hobekuntzako emaitzekin konparatuta, hauek hobetzea lortu egin da. Literaturan agertzen diren emaitzekin konparagarriak ez badira ere, proposatutako hobekuntzak emaitza itxaropentsuak ematen ditu. Gazteleratik ingeleserako esaldien itzulpenean izan ezik, beste proba guztietan emaitzak hobetzea lortu da. Honek ikerketa lerro oso interesgarria irekitzen du etorkizunean aztertzeko.

Itzultzaile automatiko neuronal bat entrenatzeko orduan, esaldi kopuru erraldoiak beharrezkoak dira, eta lan honetan erabilitakoak mugatuak izan dira. Emaitza hobeak lortuko liratezke esaldi kopuru hori handituz gero. Dataset asko eskuragarri egon badira ere, hauetatik gehienak XML formatuetan zetozen eta korrupzio arazoak zeuden, hau da, Pythonetik parser batetik pasatzerakoan, ezinezkoa zen liburutegiarentzat zuhaitza sortzea, sintaxi erroreengatik. Medline corpusean bertan ezin izan da erabili eskaintzen zen corpus osoa. Scielo corpusa erabiltzea ere saiatu da baina esandako zergatiengatik ezin izan da posiblea.

Egindako esperimentuak murritzak izan arren, emaitza itxaropentsuak ikus daitezke, gutxi bada ere, hobekuntza erakusten dutelako. Emaitza esanguratsuak lortu ahal izateko, esperimentu sakonagoak egiteaz gain, jarraian aurkeztuko diren lan ildoei heldu beharko litzaieke.

6.2 Etorkizuneko lana

Ondorioak ikusita eta lanean zehar planteatutako ideiak hartuta, etorkizuneko lan bezala hurrengoa planteatzen da:

- **Ezagutza baseko embeddingak ikasteko erlazioak berrikusi.** Emaitzetan ikusten da hobekuntzak ez duela baselinea hobetzen. Hau hobetzeko saiakera bat embeddingak entrenatzeko erlazio batzuk kentzea da. Lan honetarako ontologiako erlazio guztiak erabili dira, baina SNOMED CT begiratuta badira erlazio batzuk, lekua adierazten duena adibidez, beharbada erabilgarria ez dena. Hori dela eta, proba gehiago egin beharko lirateke atal horretan.

- **Dataset handiagoa lortu.** Ondorioetan komentatu den moduan, erabilitako data-

setaren tamaina ez da nahikoa izan. Batez ere, beharrezkoa ikusten da hurrengo lanerako domeinu biomedikoko corpus gehiago lortzea.

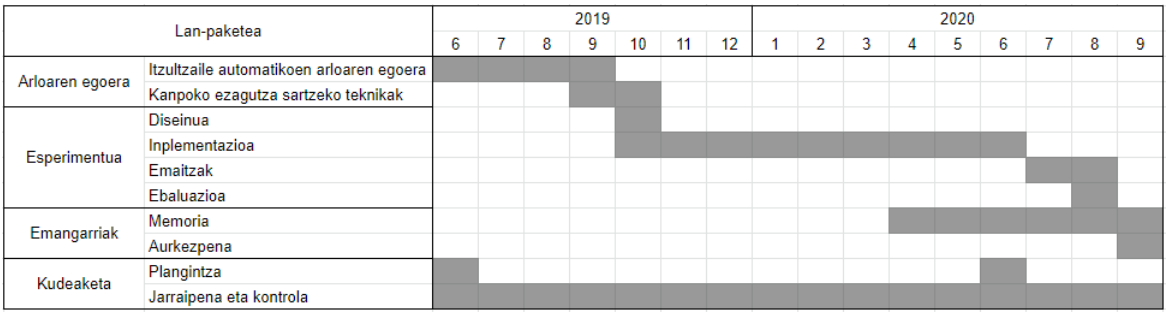
- **Parametro aurreentrenatuak erabili.** Lan honetan eredua zerotik ikasi du. Parametro aurreentrenatuak erabilita, ereduak ez du hainbeste denbora pasatu behar oinarrizko arkitektura ikasten eta Knowledge Moduluaren benetako potentziala neurtzea posiblea izango litzateke.

- **PyTorcheko Transformer inplementazioa erabili.** Lanaren garapenean zehar PyTorchek eskura jarri zuen Transformerraren inplementazioa. Lan honetako inplementazioarekin konparatuta, optimizatuagoa dator eta modulatuta etortzen denez, lan honetako hobekuntza egokitzea ez du denbora askoren beharra izango.

- **Euskarazko corpusekin trebatzea.** Lan honetan, ingelesa eta gaztelerazko corpus paraleloak erabili dira hizkuntza hauen arteko asko aurkitu daitezkeelako. Euskarazko testuak urriagoak direnez, eredua entrenatzeko zailtasun gehiago agertu daitezke. Etorkizuneko lanerako helburu bezala euskarazko testuekin trebatzea dago.

6.3 Jarraipena eta kontrola

Atal honetan proiektuaren hasierako plangintzatik abiatuta jarraipenean zehar aurkitutako arazoak eta hartutako erabakiak agertzen dira. Gainera, lan pakete bakoitzerako emandako benetako eta aurreikusitako ordu aldea komentatzen da.



6.1 Irudia: Behin betiko Gantt diagrama.

Hasteko, 6.1 irudian behin betiko Gantt diagramari begira ikus daiteke 2.2 irudian dagoen plangintzako Gantt diagramarekin alderatuta inplementazioarekin hilabete gehiagoz egon behar izan dela. Hau gertatu da zenbait atazen denbora kostua pentsatutakoa bainoa

handiagoa zelako. Horretaz gain, proiekturako eskuragarri zegoen denbora pentsatutakoa baino gutxiagoa izan da. Hori dela eta, ekainera heltzerakoan, lanaren egoera ikusita, 2.4 atalean aurreikusi zen bezala, iraileko deialdian lana aurkeztea erabaki zen.

Lan paketea	Aurreikusitako denbora	Emandako denbora
Arloaren egoera	60	65
Itzultzaile automatikoen arloaren egoera	45	45
Kanpoko ezagutza sartzeko teknikak	15	20
Esperimentua	135	172
Diseinua	10	12
Inplementazioa	110	140
Emaitzak	10	15
Ebaluazioa	5	5
Emangarriak	90	100
Memoria	80	90
Aurkezpena	10	10
Kudeaketa	15	16
Plangintza	5	5
Jarraipena eta kontrola	10	11
TOTALA	300	353

6.1 Taula: Lan pakete bakoitzaren aurreikusitako denbora eta emandako denbora.

Lan pakete bakoitzean emandako denborari begira, 6.1 taulan ikus daiteke non egon diren desbiderapen handienak. Arloaren egoeran denbora gehiago eman da ezagutza baseak eredu desberdinetan sartzeko teknikak ikertzen. Esperimentuari begira, pare bat ordu gehiago eman dira arkitekturan kanpoko ezagutza sartzeko modua diseinatzeko eta denbora gehiago eman da inplementazioan, lanaren atalik konplexuena baita eta ondo egiten dela bermatu behar delako. Emaitzak lortzeko, denbora gehiago behar izan da, batez ere, eredua entrenatzeko denbora eta entrenamenduan agertutako etenak direla eta. Amaitzeko, memoria egiteko ere denbora gehiago behar izan da. Beste lan paketeen aldetik ez da izan aldaketarik denbora gutxiko paketeak zirelako eta zerbait txarto joateko probabilitateak txikiak zirelako.

Mugarria	Data
Memoriaren entrega	2020/09/06
Aurkezpena prest izatea	2020/09/13
Lanaren defentsa	2020/09/14 - 2020/09/18

6.2 Taula: Behin betiko mugarrien datak.

Deialdi aldaketa dela eta, mugarrien daten aldaketa aldatu dira 6.2 taulan agertzen direnak behin betikoak izanik.

Bibliografia

[Bosma et al., 2009] Bosma, W., Vossen, P., Soroa, A., Rigau, G., Tesconi, M., Marchetti, A., Monachini, M., and Aliprandi, C. (2009). Kaf: a generic semantic annotation format. In *Proceedings of the GL2009 Workshop on Semantic Annotation*, pages 1–8.

[Brown et al., 1990] Brown, P. F., Cocke, J., Della Pietra, S. A., Della Pietra, V. J., Jelinek, F., Lafferty, J., Mercer, R. L., and Roossin, P. S. (1990). A statistical approach to machine translation. *Computational linguistics*, 16(2):79–85.

[Cho et al., 2014] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

[Gage, 1994] Gage, P. (1994). A new algorithm for data compression. *C Users Journal*, 12(2):23–38.

[Goikoetxea et al., 2015] Goikoetxea, J., Soroa, A., and Agirre, E. (2015). Random walks and neural network language models on knowledge bases. In *Proceedings of the 2015 conference of the North American Chapter of the Association for Computational Linguistics: Human language technologies*, pages 1434–1439.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

[Hutchins and Somers, 1992] Hutchins, W. J. and Somers, H. L. (1992). *An introduction to machine translation*, volume 362. Academic Press London.

[Joulin et al., 2016] Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., and Mikolov, T. (2016). Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

[Kalchbrenner and Blunsom, 2013] Kalchbrenner, N. and Blunsom, P. (2013). Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709.

[Kirchhoff and Yang, 2005] Kirchhoff, K. and Yang, M. (2005). Improved language modeling for statistical machine translation. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, pages 125–128. Association for Computational Linguistics.

[Koehn, 2005] Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer.

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

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

[Mikolov et al., 2013] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[Miller, 1998] Miller, G. A. (1998). *WordNet: An electronic lexical database*. MIT press.

[Oronoz et al., 2013] Oronoz, M., Casillas, A., Gojenola, K., and Perez, A. (2013). Automatic annotation of medical records in spanish with disease, drug and substance names. In *Iberoamerican Congress on Pattern Recognition*, pages 536–543. Springer.

[Papineni et al., 2002] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

[Pennington et al., 2014] Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

[Sennrich et al., 2015] Sennrich, R., Haddow, B., and Birch, A. (2015). Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

[Shi et al., 2018] Shi, T., Keneshloo, Y., Ramakrishnan, N., and Reddy, C. K. (2018). Neural abstractive text summarization with sequence-to-sequence models. *arXiv preprint arXiv:1812.02303*.

[Simard and Plamondon, 1998] Simard, M. and Plamondon, P. (1998). Bilingual sentence alignment: Balancing robustness and accuracy. *Machine Translation*, 13(1):59–80.

[Srivastava et al., 2014] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

[Sutskever et al., 2014] Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

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

[Weaver, 1955] Weaver, W. (1955). Translation. *Machine translation of languages*, 14:15–23.

[Yang and Mitchell, 2019] Yang, B. and Mitchell, T. (2019). Leveraging knowledge bases in lstms for improving machine reading. *arXiv preprint arXiv:1902.09091*.