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

Informatika Ingeniaritzako Gradua

Konputazioa

Gradu Amaierako Lana

Embedding eleartekoak sortzeko mapaketa metodoen mugen azterketa

Egilea

Aitor Ormazabal Oregi

2019

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

Informatika Ingeniaritzako Gradua

Konputazioa

Gradu Amaierako Lana

Embedding eleartekoak sortzeko mapaketa metodoen mugen azterketa

Egilea

Aitor Ormazabal Oregi

Zuzendaria(k)

Eneko Agirre eta Aitor Soroa

Laburpena

Hitz-embeddingak bokabulario bateko hitzen eta bektore espazio baten arteko mapaketak dira. Embedding horiek hitzak bektore espazio bateko puntu bezala adieraztea ahalbidetzen digute, eta hizkuntza prozesamenduko arloko hainbat atazatan arrakasta handia izan dute. Hitz-embedding elebidunak bi hizkuntzetako hitzak bektore espazio berdinera mapatzen dituzten embeddingak dira. Embedding horiek sortzeko metodoak bi klasetan sailka daitezke: mapaketa metodoak eta aldibereko metodoak. Mapaketa metodoek hizkuntza bakoitzeko embeddingak independenteki sortu eta ondoren transformazio linealen bidez espazio amankomun batera mapatzen dituzte. Aldibereko metodoek, berriz, zuzenean espazio amankomunan ikasten dituzte bi hizkuntzetako bektoreak.

Azken urteetan embedding elebidunen inguruko ikerkuntza mapaketa metodoetara bideratuta egon da. Metodo horiek gainbegiratze maila oso txikia eskatzen dute, eta corpus elebakarrekin entrenatu daitezke; ondorioz, arrakasta handia izan dute aplikazio praktikoetan. Dena den, badituzte arazoak.

Arazo horietako bat hubness delakoa da. Hubnessak eragiten du dimentsio altuko espazioetan puntu gutxi batzuk beste puntu askoren gertukoen auzokideak izatea, eta embedding elebidunen kalitatean eragin negatiboa du. Lan honetan fenomeno honen hainbat iturri posible proposatzen ditugu, eta bakoitzaren eragina neurtzen dugu.

Gainera, mapaketa metodoen erabilera justifikatzeko isomorfismo hipotesia erabili izan da, hizkuntza desberdinetako embeddingek egitura antzekoa dutela esaten duena. Hainbat autorek hipotesi hori zalantzan jarri dute, eta frogatu dute betetzen ez denean mapaketa metodoek ez dutela ondo funtzionatzen. Hala ere, ez dago argi ea egituren desberdintasun hori mapaketa metodoen muga bat den, edo embedding elebidunak ikastean agertzen den arazo orokorrago bat den. Hori aztertzeko, corpus paraleloak erabiliz mapaketa metodoak eta aldibereko metodoak alderatzen ditugu. Frogaten dugu, baldintza ideal hauetan, aldibereko ikasketa metodoen bidez sortatuko embeddingak hobeto lerrokatzen direla,

hubness txikiagoa dutela, eta hiztegi indukzioan errendimendu hobea dutela. Halaber, ondorioztatzen dugu egungo mapaketa metodoek muga larriak dituztela, eta ikerkuntza lerro interesgarria izan daitekela seinale elebidun ahulago batekin embedding elebidunak ikasteko aldibereko teknikak aztertzea.

Eskerrak eman nahi dizkiet Eneko Agirre eta Aitor Soroa zuzendariei proiektua ezin hobeki bideratzeagatik, eta Mikel Artetxe eta Gorka Labaka IXA taldeko kideei, lan hau burutzeko eskaini didaten laguntza baliotsuagatik.

Gaien aurkibidea

Irudien aurkibidea

Taulen aurkibidea

1. KAPITULUA

Sarrera

Hitz-embeddingak bokabulario bateko hitzen eta bektore espazio baten arteko mapaketak dira. Embeddingek hitzak bektore espazio bateko puntu bezala adieraztea ahalbidetzen digute, eta puntu horiek hizkuntzaren mapa semantiko moduko bat osatzen dute; espazio horretan bektoreen arteko distantziak eta erlazioak semantikoki esanguratsuak izango dira. Bektore-adierazpen hauek hizkuntza prozesamendu eta ikasketa sakoneko arloetan oso arrakastasuak izan dira, eta hainbat atazen errendimendua hobetu dute; besteak beste, sentimendu analisian, analisi sintaktikoan, eta itzulpen automatikoan artearen egoerako metodoek hitz-embeddingak erabiltzen dituzte.

Hitz-embedding elebidunak embedding mota berezi bat dira, non bi hizkuntzatako hitzak espazio amankomun berdinera mapatzen diren, koherentzia semantikoa mantenduz. Embedding elebidunek hainbat erabilera dituzte; besteak beste, itzulpen automatiko ez-gainbegiratuan, hiztegi indukzio elebidunean eta transferentzia-ikasketan oso erabiliak dira.

Azken urteetan hitz-embedding elebidunen inguruko ikerkuntza “offline” mapaketa-metodoetara bideratuta egon da. Metodo horiek hizkuntza bakoitzeko embeddingak independenteki ikasten dituzte, eta ondoren transformazio linealak erabiltzen dituzte bi hizkuntzetako hitz-bektoreak espazio amankomun batera mapatzeko. Mapaketa-metodoak oso arrakastatsuak izan dira, gainbegiratze elebidun oso txikia eskatzen dutelako. Hala ere, badituzte arazoak.

Arazo hauetako bat hubness deritzona da. Fenomeno honek eragiten du dimentsio altuko espazioetan puntu gutxi batzuk beste puntu askoren gertukoen auzokideak izatea, eta

hitz-embedding elebidunen kalitatean eragin negatiboa du. 4. kapituluan hitz-embedding elebidunetan agertzen den hubnessa aztertzen dugu; haren iturri izan daitezkeen hainbat faktore proposatzen ditugu, eta bakoitzaren eragina neurtzen dugu. Halaber, hubnessa arintzeko proposatu zen CSLS neurriaren eraginkortasuna aztertzen dugu.

Mapaketa-metodoen erabilera justifikatzeko isomorfismo hipotesia erabili izan da, hizkuntza desberdinetako embeddingek egitura antzekoa dutela esaten duena. Hipotesi honen arabera, hizkuntza desberdinen embeddingak independenteki ikastean puntu-espazio berdina lortzen da, baina modu desberdinean orientatua. Hori betetzen bada, posible izango da transformazio lineal baten bidez bi espazioak berriro lerrokatzea, eta ondorioz mapaketa-metodoak arrakastatsuak izango dira.

Hala ere, hainbat autorek hipotesi hori zalantzan jarri dute (Søgaard et al. (2015)), frogatuz hizkuntza desberdinetako embeddingak independenteki ikastean espazioen egitura geometrikoan desberdintasun handiak agertzen direla. Baina ez dago argi ea egituren arteko desberdintasun hori embeddingak independenteki ikastearen ondorio bat den, edo dibergentzia linguistikoen ondorioz embedding elebidunak ikastean agertzen den arazo orokorragoa den.

Galdera horri erantzuna emateko, mapaketa-metodoez gain embeddingak ikasteko aldibereko metodo bat aztertzen dugu. Metodo honek hizkuntza bakoitzeko embeddingak independenteki ikasi eta ondoren transformazio linealen bidez mapatu ordez zuzenean embeddingak espazio amankomunan aldi berean sortzen ditu.

5. kapituluan mapaketa-metodoak eta aldibereko metodoak alderatzen ditugu. Kapitulu honetako esperimentuen bidez aztertu ahal izango dugu ea isomorfia-eza eta hubness fenomeno patologikoak embeddingak independenteki ikasi eta transformazio linealen bidez mapatzearen ondorioz agertzen diren, edo embedding elebidunen berezko arazoak diren.

Azkenik, 6. kapituluan ateratako ondorioak laburtzen ditugu, eta ondorio horietan oinarrituta etorkizunerako ikerkuntza-lerro bat proposatzen dugu.

Lan hau gradu amaierako lan baten irismenetik at joan da, eta hizkuntza prozesamenduko arloan oso erabiliak diren hitz-embeddingen inguruan ekarpen zientifiko bat egin da. Lan honetako edukian oinarritutako artikulo bat ACL 2019[1] konferentzian argitaratuko da.

[1] ACL SCIE Class 1 konferentzia bat da. http://www.acl2019.org/EN/index.xhtml

2. KAPITULUA

Proiektuaren Helburuen Dokumentua

Kapitulu honetan, proiektuaren deskribapena eta honen helburuak azaltzeaz gain, lanaren deskonposaketa egitura (LDE) eta landuko diren atazen zerrenda emango dira. Honez gain, barne- eta kanpo-mugarriak, aurreikusitako dedikazioaren zenbatespena eta Gantt diagrama azalduko dira. Gainera, arrisku plana zehaztuko da eta proiektuaren jarraipen eta kontrola burutuko da.

2.1 Proiektuaren deskribapena eta helburua

Proiektu hau ikerkuntza proiektua izango da, zeinen helburua hitz-embedding elebidunetan agertzen den hubness arazoaren iturriak, eta, posible izatekotan, konponbide bat aurkitzea izango baita. 3 kapituluan ikusiko den bezala, hitz-embedding elebidunak bi hizkuntzetako hitzen eta bektore espazio amankomun baten arteko mapaketak dira, eta hitzak bektore bezala adieraztea baimentzen digute. Bektore-adierazpen hauek hitzei buruzko informazio semantikoa jasotzen dute, eta oso erabilgarriak dira aplikazio praktikoetan. Hubnessa dimentsio altuko espazioetan agertzen den fenomenoa da, puntu gutxi batzuk beste puntu askoren gertukoen auzokideak izatea eragiten duena. Gainera, ikusiko da hubnessak eragin negatiboa duela embedding elebidunen kalitatean, eta ondorioz interesgarria da fenomeno honen iturriak hobeto ulertzea. Literaturan dimentsio altuko espazioen testuinguruan hubnessa fenomeno ezaguna da (Radovanović et al., 2010b,a), eta hitz-embedding elebidunen kasuan agertzen dela ere badakigu (Lazaridou et al., 2015; Shigeto et al., 2015; Conneau et al., 2018)), baina lan hau hastearen datan oraindik ez

da egon hitz-embedding elebidunen kasuan hubnessa eragiten duten faktore desberdinak sistematikoki aztertzen dituen lanik.

Lan honetan faktore hauek aztertuko dira, hubness fenomenoa hobeto ulertzeko, eta ahal den heinean arintzeko helburuarekin. Horretarako lehenik artearen egoera aztertu beharko da, eta ondoren esperimentazioa diseinatu eta inplementatu beharko da. Azkenik, esperimentuaren emaitzak aztertu eta ondorioak erauzi beharko dira.

Gainera, emaitza positiboak lortzekotan lan honen helburuetako bat konferentzia batean ikerkuntza artikulo bat argitaratzea izango da. Honek gradu amaierako lanaren mugarrietaz gain beste mugarri bat gehituko du (aukeratutako konferentziaren epemuga, alegia).

2.2 Proiektuaren plangintza

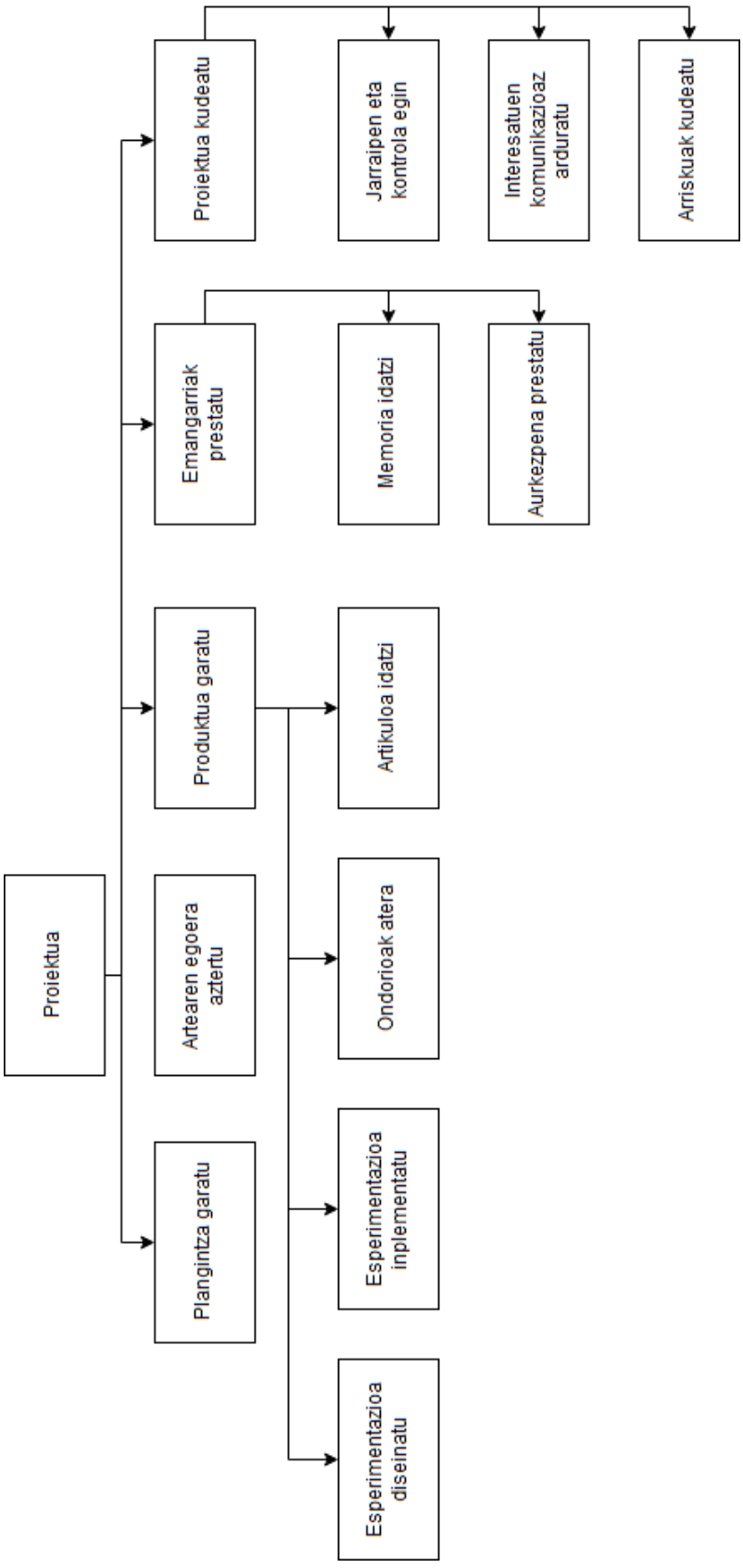
2.2.1 Lanaren antolaketa

Atal honetan proiektua burutzeko beharrezko atazak eta plangintzan bakoitzari esleitutako denbora azaltzen dira. Lehenik lanaren deskonposaketa eta ataza bakoitzaren azalpena emango dugu, eta ondoren ataza hauek biltzen dituen kronograma aurkeztuko dugu.

Lanaren deskonposaketa egitura (LDE)

2.1 irudian lanaren deskonposaketa hierarkikoa aurkezten da. Identifikatutako atazak eta azpiatazak hurrengoak dira:

- **Plangintza garatu.** Ataza honetan proiektua burutzeko erreferentziatzat hartuko den plangintza garatuko da. Plangintzan proiektuaren atazak, denboraren banaketa, emangarriak eta mugarriak definituko dira. Gainera, komunikazio- eta arriskuplanak zehaztuko dira.
- **Artearen egoera aztertu.** Ikasketa pauso honetan hitz-embedding elebidun eta hubnessaren inguruko artearen egoera aztertuko da. Artearen egoera ezagutzea ezinbestekoa izango da, alde batetik egingo den lana dagoeneko eginda ez dagoela ziurtatzeko, eta bestetik egungo metodo onenak kontuan hartzen dituen kalitatezko esperimentazioa diseinatu ahal izateko.
- **Produktua garatu.** Ataza honetan ikerkuntza proiektua burutuko da.



2.1 Irudia: Proiektuko Lanaren Deskonposaketa Egitura diagrama.

  - **Esperimentazioa diseinatu.** Azpiataza honen eginkizuna esperimentua inplementatu ahal izateko beharrezkoak diren faktore desberdinak zehaztea izango da.
  - **Esperimentazioa inplementatu.** Azpiataza honetan esperimentazioa kodean inplementatu eta egikaritu egingo da.
  - **Ondorioak atera.** Azpiataza honen helburua egindako esperimentuaren emaitzak aztertzea, interpretatzea eta ondorioak erauztea izango da.
  - **Artikuloa idatzi.** Azken azpiataza honetan aurrekoetan egindako lana eta ateratako ondorioak ikerkuntza artikulo batean jasoko dira. Artikulo honen formatua aukeratutako konferentziaren gidalerroekin bat etorri beharko da.

- **Emangarriak prestatu.** Ataza hau gradu amaierako lanerako eskatzen diren emangarriak garatzean datza. Emangarri hauek memoria eta aurkezpena dira, eta ondorioz ataza hau bi azpiatazetan banatzen da:
  - **Memoria idatzi.**
  - **Aurkezpena prestatu.**

- **Proiektua kudeatu.** Ataza honen helburua proiektuaren garapen zuzena bermatzea izango da, hau da, ziurtatzea garapenak plangintzan zehaztutako bidea jarraitzen duela, eta, hau ezinezkoa edo desegokia denean, plangintza egokitzea. Honetarako arrisku- eta komunikazio-planak zehaztuko dira, eta proiektuaren jarraipen eta kontrola burutuko da. Hau da, ataza hau hurrengo azpiatazetan banatuko da:
  - **Jarraipena eta kontrola egin.**
  - **Interesatuekiko komunikazioaz arduratu**.
  - **Arriskuak kudeatu.**

Mugarriak

Proiektuak hainbat kanpo- eta barne-mugarri edukiko ditu. Alde batetik, proiektuak bete behar dituen kanpo-mugarriak hurrengoak dira:

- **2019-03-04:** ACL konferentziako artikuloak bidaltzeko epemuga.
- **2019-06-23:** Memoria entregatzeko azken data.

Ataza	Aurreikusitako orduak
Plangintza garatu	5
Artearen egoera aztertu	80
Esperimentazioa diseinatu	10
Esperimentazioa inplementatu	115
Ondorioak atera	10
Artikuloa idatzi	15
Memoria idatzi	60
Aurkezpena prestatu	5
Guztira	300

2.1 Taula: Ataza bakoitzerako aurreikusitako ordu kopuruak.

- **2019-07-01 eta 2019-07-12 artean:** Defentsa egin.

Hauez gain, hurrengo barne-mugarriak zehaztu ditugu:

- **2019-02-04:** Esperimentuak bukatu, eta erabaki ea merezi duen artikuloa idaztea.
- **2019-06-14:** Memoria bukatu zuzendariek berrikusteko.

Kronograma eta aurreikusitako ordu kopuruak

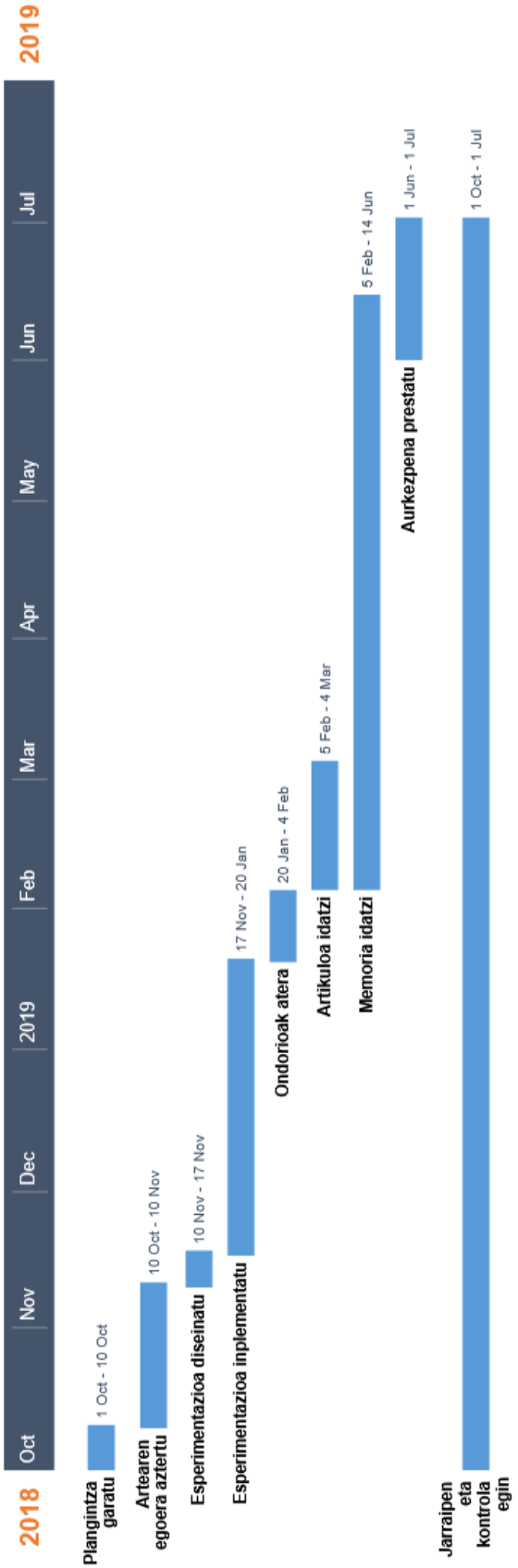
2.2 irudian aurreko atalean azaldutako ataza desberdinei esleitutako epeekin osatutako kronograma edo Gantt diagrama aurkezten da, eta 2.1 taulan ataza bakoitzarentzat aurreikusitako ordu kopuruak ikusi daitezke.

2.2.2 Komunikazio-plana

Atal honetan proiektuko komunikazio-plana azaltzen da. Plan honek proiektuaren interesatuak, haien arteko komunikazio-kanalak, eta azkenik proiektuko informazio-sistema zehaztuko ditu.

Interesatuen identifikazioa

Proiektuaren helburua eta aukeratutako lan-lerroa kontuan hartuz, hurrengo interesatuak identifikatu dira:



2.2 Irudia: Proiektuko kronograma

- **Aitor Ormazabal**. Interesatu nagusia proiektua bera burutu behar duen Aitor Ormazabal izango da, Informatika Ingeniaritzako Graduko ikaslea. Alde batetik, ikasketaplanean gradu amaierako lanari derrigorrezko 12 kreditu dagozkio, eta ondorioz ezinbestekoa da gradua bukatu ahal izateko. Gainera, jorratutako gaia ikaslearentzat interes handikoa izateaz gain, ikerkuntza munduari buruz ikasteko aukera paregabea da.

- **Aitor Soroa eta Eneko Agirre**. Aitor Soroa eta Eneko Agirre, biak IXA taldeko kideak, Informatikan Doktoreak, eta EHUko Informatika Fakultateko irakasleak, proiektuko zuzendariak izango dira. Proiektuaren ikuskapenaz arduratuko dira, garapena gainbegiratuz eta esperimentuen norabideari buruzko erabakietan parte hartuz.

- **Mikel Artetxe**. Mikel Artetxe EHUn Informatika Doktoregaia eta ikerlaria da, Hizkuntzaren Prozesamenduaren (*Natural Language Processing* edo NLP ingelesez) arloan aritzen dena. Mikelen ikerkuntza-lerroa proiektuaren esparruarekin lotuta dago, eta ondorioz bere parte-hartzea oso lagungarria izango da. Proiektuaren eta esperimentuen norabideari buruzko erabakietan parte hartuko du.

- **Gorka Labaka**. Gorka Labaka Informatikan doktorea eta IXA taldeko ikerlaria da, itzulpen automatikoaren arloan diharduena. Proiektuaren eta esperimentuen norabideari buruzko erabakietan parte hartuko du.

Lan-metodologia eta komunikazio-kanalak

Aitor Ormazabal ikasleari IXA taldeko zerbitzarietarako atzipena baimendu zaio, esperimentuak burutzeko beharrezkoak baina konputazionalki garestiak izango diren kalkuloak egikaritu ahal izateko. Ikasleak lana etxetik eta baita ere fakultatean egingo du, IXA taldeak lan-poltsa baten harira informatika fakultateko bulego batean esleitutako mahai batean.

Proiektuko interesatu desberdinen arteko komunikazio kanalak bi izango dira:

- **Bilerak.** Eskuarki astero egingo dira bilerak Eneko Agirre eta Aitor Soroa zuzendariekin, Enekoren bulegoan. Bilera hauetan proiektuaren jarraipena egingo da, non Aitor Ormazabalek zuzendariei proiektuaren egoeraren berri emango dien. Bilera hauetan ere proiektuaren eta esperimentuen norabideari buruzko erabakiak hartuko dira. Alderdi teknikoei buruz erabaki garrantzitsuak hartzeko orduan, edo beste

arrazoiren batengatik haientzat interesgarria izan daitekeenean, Mikel Artetxe eta Gorka Labakak ere parte hartuko dute aurrez-aurreko bilera hauetan.

- **Posta elektronikoa.** Posta elektronikoa erabiliko da kideen artean komunikatzeko, bileren datak zehazteko, garrantzi txikiko kontuak argitzeko, edo bilera fisikoek baimentzen dutena baino maiztasun handiagoko komunikazioa beharrezkoa denean (adibidez konferentziaren epemuga gertu dagoenean).

Informazio-sistema

Proiektuko eduki guztiak era digitalean gordeko dira. Erabiliko diren datu gordelekuak hurrengoak izango dira:

- **IXA taldearen zerbitzariak.** Kostu konputazional handiko kalkuloak edozein modutan zerbitzari hauetan egikaritu beharko direnez, erabaki da garapena zuzenean zerbitzari hauetan egitea. Zerbitzariak SSH bidez bulegotik edo etxetik atzitu ahal izango dira, eta VIM edo EMACS moduko erreminten bidez egin ahal izango da garapena. Gainera, zerbitzari hauek segurtasun-kopiak dituzte, beraz datuak galtzeko aukera oso txikia izango da.

- **Etxeko ordenagailua.** Garapena gehienbat IXAko zerbitzarietan egingo den arren, zenbait kasutan ere lokalki egingo da (adibidez grafikak sortu behar direnean, SSH konexio bidez hau egitea deserosoa baita). Kasu hauetan urrutiko zerbitzarietatik atzituko dira garapena egiteko beharrezko datuak saio bakoitzean, eta bukatzean berriro zerbitzarira igoko dira, hau da, ez dira lokalki mantenduko. Izan ere, etxeko ordenagailuan ez dago segurtasun kopiarik, eta ondorioz datuak galtzeko aukera askoz altuagoa da.

- **Google Drive.** Taldearentzat barne-txostenak egiten direnean Google Driven sortuko dira. Zerbitzu hau erabiltzea erabaki da banatzeko oso erraza delako (nahikoa da web arakatzaile bat eta interneterako konexio bat), eta taldekide guztiek ezagutzen dutelako.

- **Overleaf.** Emangarrietarako kalitate handiko dokumentuak sortu behar direnean (adibidez memoria edo artikuloa) LaTeX teknologia eta Overleaf zerbitzua erabiliko dira. Zerbitzu hau aukeratu da Google Driven antzera lana modu errazean banatzea ahalbidetzen duelako.

2.2.3 Arrisku-plana

Atal honetan proiektuko arrisku-plana azaltzen da. Horretarako arrisku posible desberdinak eta bakoitzaren deskribapena, inpaktua eta mitigazio-neurriak edo kontingentzia-planak zehazten dira.

Proiektuaren natura eta zehaztutako informazio-sistema kontuan hartuz, bi arrisku nagusi identifikatu ditugu: informazio-galera eta esperimentuek emaitza negatiboak ematea.

Informazio-galera

- **Deskribapena:** Posible da arazo teknikoengatik Google Driven, Overleafen, edo IXA zerbitzarietan gordetako lanaren galera bat sufritzea.
- **Probabilitatea:** Oso txikia.
- **Kontingentzia plana:** Esan dugun bezala, datuak gordetzeko erabilitako hiru zerbitzuek dagoeneko kontingentzia planak (segurtasun kopiak) erabiltzen dituzte datuen galerak ekiditeko. Hala ere datuen galera bat egoten bada, konturatu bezain laster aztertuko da zein datu galdu diren, eta gordetako azken puntutik hasiko da lana. Galdutako edukia bizkor berregitea ezinezkoa bada plangintza egokitu beharko da.

Esperimentazioan emaitza negatiboak lortzea

- **Deskribapena:** Posible da esperimentazioak emandako emaitza ez izatea esperotakoak, edo ondorio argi bat ezin ateratzea.
- **Probabilitatea:** Altua.
- **Mitigazio-neurriak:** Lortutako emaitzak ez badira positiboak seguraski ez dira argitaratzeko modukoak izango, baina hala ere memorian lortutako emaitzak eta atera ahal izan diren ondorioak jasoko dira.
- **Kontingentzia-plana:** Kronograman esperimentuen emaitza lortzeko epea konferentziaren epemuga baino hilabete bat lehenago jarri da, eta printzipioz hilabete hori artikuloaren idazketa egiteko da. Hala ere, posible da esperimentuen emaitzak oso argiak ez izatea, baina esperimentazio edo lan gehigarri baten bidez emaitza indartsuagoak lortzeko aukera izatea. Hau gertatzekotan, hilabete horretan esperimentazio gehigarri hau diseinatu eta egikaritu daiteke, eta ondoren berriro ebaluatu ea emaitza berriekin argitaratu daitekeen.

2.2.4 Jarraipen eta kontrola

Atal honetan lanaren jarraipenean zehar aurkitutako arazoak eta hartutako neurriak azalduko ditugu. Gainera, aurreikusitako denboren eta proiektuan zehar benetan ataza bakoitza egiteko erabilitako denboraren arteko alderaketa egingo dugu.

Aurkitutako arazoak eta plangintzaren egokitzapenak

Lanean zehar aurkitutako arazo garrantzitsu bakarra lehen esperimentazioan emaitza ez oso argiak lortu genituela izan da, arrisku-planan aurreikusi zen bezala. Izan ere, hubness fenomenoari buruzko iturriak aztertzeko esperimentazioaren emaitzetan metodoen arteko desberdintasun nabariak ez dira ikusi, eta argi ateratako emaitzak bi izan dira bakarrik: domeinu eta hizkuntza desberdintasunak hubnessa sortzen dutela, eta aldibereko metodoak mapaketa-metodoek baino propietate hobek dituela. Domeinu eta hizkuntzari buruzko emaitzak argiak dira, baina ez zitzaizkigun argitaratzeko moduko nahikoak iruditu. Hala ere, pentsatu genuen aldibereko metodo eta mapaketa-metodoen arteko alderaketan sakontzea, hortik bide interesgarria ikusi genuelako. Ondorioz, arrisku-planan aurreikusi genuen bezala esperimentazio gehigarri bat diseinatu eta egikaritu zen, eta honen ondoren berriro ebaluatu zen ea artikuloa idaztea merezi zuen, eta baietz erabaki zen. Gainera, arrisku-planan zehaztu bezala hasierako esperimentuak eta emaitzak, naiz eta artikuloan ez jaso, memorian bai aurkeztu dira.

Plangintzan egin behar izan den egokitzapen bakarra, arrisku-planan aurreikusi den bezala, lehen esperimentazioaren bukaeraren eta konferentziako epemugaren arteko hilabetean bakarrik artikuloa idatzi ordez esperimentazio gehigarria egikaritu eta ondoren artikuloa idatzi behar izan dela da. Gainerako atazen burutzea eta kronograma berdin-berdina izan da. Gainera, honen ondorioz esperimentazioa burutzearen atazak behar izan duen denbora hasiera batean aurreikusitakoa baino nahiko gehiago izan da.

Aurreikusitako eta benetako denbora dedikazioen alderaketa

2.1 irudian ataza bakoitzarentzat aurreikusitako ordu kopuruak adierazi ditugu, eta orain 2.2 irudian benetan proiektuan zehar erabilitako ordu kopuruak aurkezten dira. Aurreko atalean esan bezala, aurreikusitako eta benetako kopuruen arteko desberdintasun nagusia esperimentazioaren diseinu eta inplementazioaren atazetan dago, esperimentazio gehigarri bat diseinatu behar izan delako.

Ataza	Erabilitako orduak
Plangintza garatu	5
Artearen egoera aztertu	83
Esperimentazioa diseinatu	16
Esperimentazioa inplementatu	152
Ondorioak atera	9
Artikuloa idatzi	16
Memoria idatzi	73
Aurkezpena prestatu	6
Guztira	360

2.2 Taula: Ataza bakoitzerako benetan erabilitako orduak

3. KAPITULUA

Aurrekariak

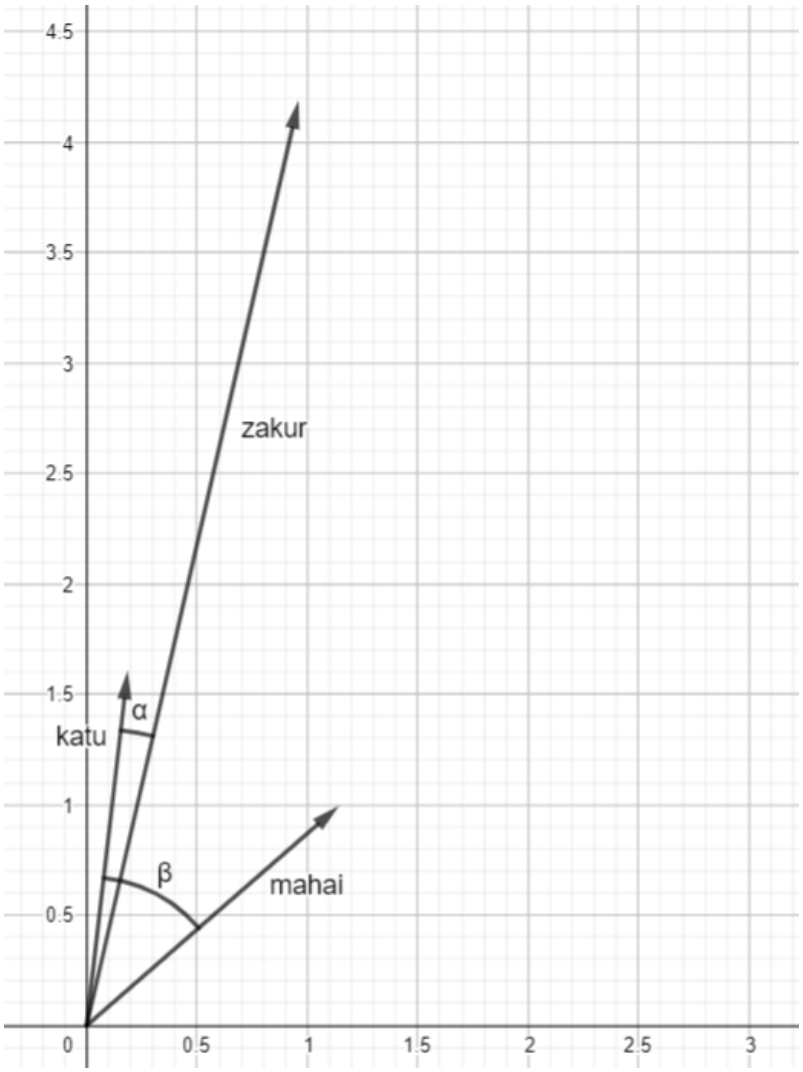
3.1 Hitz-embeddingak

Hitz-embeddingak [1] hizkuntza bateko edo gehiagoko hitzen eta $\mathbb{R}^N$ bektore espazio bateko elementuen arteko mapaketak dira, hau da, $\Phi : B \to \mathbb{R}^N$ motako mapaketak, non B bokabulario edo hitz multzo bat den. Hitz-embedding baten bitartez hitz bakoitza bektore numeriko baten bidez adierazten da. Mapaketa horiek lortzeko modu asko daude; horietako bat, oso sinplea, **one-hot** adierazpena deritzona da. Metodo honetan, V hitzez osatutako bokabulario bat badaukagu, $B = \{w_1, ..., w_V\}$, non w_i-ek hitzak adierazten dituzten, orduan V dimentsioko bektoreak erabiltzen dira hitzak errepresentatzeko. Zehazki, $\Phi : B \to \mathbb{R}^V$ mapaketak definituko du embeddinga, non i bakoitzeko $\Phi(w_i) = (\delta_{1i}, ..., \delta_{Vi})$ (hemen δ_{ij} kronecker delta da), hau da, w_i hitzaren bektore adierazpenak 1 zenbakia edukiko du i. koordenatuan, eta 0 beste guztietan.

Hala ere, deskribatutako one-hot hitz-embeddinga ez dirudi oso erabilgarria; bektore bakoitzak, dagokion hitza modu unibokoan identifikatzen duen arren, ez digu hitzari buruzko inongo informaziorik ematen. Horregatik, normalean ez da hitz-embedding terminoa erabiltzen one-hot adierazpenak deskribatzeko. Izan ere, normalean hitz-embeddingetaz ari garenean espero dugu mapaketak bi propietate betetzea:

1. **Banatua** izan behar da, hau da, mapaketak bokabularioko hitzak bektore espaziotik

[1] *embedding* terminoa oso zabaldua dago ingelesezko literaturan, eta ez dugu euskaratzeko termino argi bat aurkitu, beraz mailegutzat hartu dugu testuan zehar.



3.1 Irudia: Bektoreen arteko angeluak erabiltzen dira antzekotasunak neurtzeko

banatuko ditu. Adibidez, one-hot adierazpenak ez du propietate hau betetzen, bektoreak $\mathbb{R}^N$-ren $A = \{x \in \mathbb{N}^N : \sum_{n=1}^{N} x_n = 1\} \in \mathbb{N}^N$ azpimultzoan bakarrik kokatzen baitira.

2. Hitz baten bektoreak horri buruzko nolabaiteko **informazio semantikoa** jaso behar du.

Zer esan nahi du bigarren propietateak? Ikusiko ditugun algoritmoen bidez hitz-embeddingak sortu ondoren, bektoreen arteko distantziak eta erlazioak semantikoki esanguratsuak izango dira. Adibidez, ikusiko dugu bektoreen arteko distantzia eta dagokien hitzen ahaidetasun semantikoa erlazionatuak daudela. Normalean, hitz-embeddingen kasuan bektoreen arteko distantzia neurtzeko kosinu-antzekotasuna erabiltzen da, hau da, bektoreen

arteko angeluaren kosinua (balio hori ez da distantzia neurria, antzekotasun neurria baizik, balio altuagoek bektoreak gertuago daudela adierazten duelako). Hau egiten da hitz-embeddingen kasuan bektoreen arteko angeluak luzera-desberdintasunak baino esanguratsuagoak direla ikusi delako. Distantzia euklidear arrunta eta kosinu-antzekotasunaren arteko desberdintasuna 3.1 irudian ikusi daiteke. Irudiko kasuan, distantzia euklidear arrunta erabiltzen bada *katu* eta *mahai* hitzen bektoreak *katu* eta *zakur* hitzenak baino gertuago daude, baina angeluen arabera *katu* eta *zakur* hitzen bektoreak askoz gertuago daude. Hitz-bektoreekin lan egitean, emaitza hobeak lortzen dira kosinu-antzekotasuna erabiltzean.

Beraz, ondo entrenatutako euskarazko hitz-embedding batean x, y eta z *katu, zakur* eta *mahai* hitzen bektoreak badira, $\cos(x,y) > \cos(x,z)$ eta $\cos(x,y) > \cos(y,z)$ izatea espero dezakegu. Bi $x, y \in \mathbb{R}^N$ bektoreen arteko kosinua honela definitzen da:

$$\cos(x,y) = \frac{x \cdot y}{\|x\|\|y\|} = \frac{\sum_{j=1}^{n} x_j y_j}{\sqrt{\sum_{j=1}^{n} x_j^2}\sqrt{\sum_{j=1}^{n} y_j^2}}.$$

Propietate horiek betetzen dituzten kalitatezko hitz-embeddingak sortzearen inguruan ikerkuntza ugaria egon da, eta hainbat metodo existitzen dira. Metodo horien baliozkotasuna orokorrean hipotesi distribuzionala delakoaren mendekoa da. Hipotesi horrek testuinguru antzekoetan agertzen diren hitzak esanahi antzekoa izateko joera dutela dio. Hori jakinda, pentsa dezakegu testu corpus handi batetik hitzen agerkidetza informazioa (hau da, zenbat aldiz hitz pare desberdinak testuinguru berdinetan agertzen diren) erauzi eta erabili dezakegula hitz-bektoreak lortzeko. Hau egiteko hainbat metodo proposatu dira: kontaketan oinarritutakoak, sare neuronaletan oinarritutakoak, eta beste asko. Gu sare neuronaletan oinarritutakoetan zentratuko gara, zehazki Word2Vec metodoan.

3.1.1 Word2Vec

Word2Vec hitz-embeddingak sortzeko algoritmo familia bat da, Mikolov et al. (2013b) lanean proposatua. Familia horrek algoritmo berdinaren bi bertsio jasotzen ditu, **CBOW** (**Continuous Bag of Words**) eta **Skip-gram** izenekoak. Gainera, ikusiko dugun bezala bertsio bakoitzak bi aldaera ditu. Guk lehenik CBOW bertsioa deskribatuko dugu.

Hitz-embeddingak lortzeko metodo honetan lehenik sare neuronal bat entrenatzen da ataza lagungarri bat betetzeko, eta bigarrenik entrenatutako sareko pisuetatik erauzten dira hitz-bektoreak. Atal honetan algoritmoaren deskribapen zehatza emango dugu.

Sare neuronala entrenatzeko erabiltzen den ataza lagungarria ondorengoa da: hitz baten testuingurua emanda hitza aurreikusten ikastea. Horretarako geruza ezkutu bakarreko sare arrunt bat erabiltzen da, ezkutuko geruzan ez duena aktibazio ez-linealik erabiltzen, eta irteera geruzan softmaxa erabiltzen duena. Hori formalizatzeko, demagun V hitz desberdineko bokabulario bat dugula, eta eta N dimentsioko hitz-bektoreak sortu nahi ditugula gure embeddingean. Orduan, sareak bi pisu matrize edukiko ditu, W_1 eta W_2. W_1 matrizeak sarrera geruza eta geruza ezkutuko arteko pisuak gordeko ditu, eta $V \times N$ tamainakoa izango da. Modu berean W_2-k geruza ezkutu eta irteera geruza arteko pisuak gordeko ditu, eta $N \times V$ tamainakoa izango da. Sarrerako testuingurua $N \times 1$ tamainako $x = (x_1, ..., x_V)$ bektore baten bidez adieraziko da, non x testuinguruko hitz guztien one-hot adierazpenen bataz bestekoa den. Gauzak horrela, sarearen irteera $y = softmax(W_2W_1x)$ probabilitate bektorea izango da, non y-ko i. elementua aurreikusten ari garen hitza bokabularioko i. izatearen probabilitatea izango den. Hau da, sareak ezagutzen ez dugun hitz baten h testuingurua jaso eta hitz hori w_i izatearen probabilitatea emango digu. Probabilitate hori $P(w_i|h)$ bezala adieraziko dugu.

Behin eredua definituta, sarea entrenatzeko $C = \{w_1, ..., w_K\}$ corpus bat erabiltzen da. Lehenik c leiho tamaina aukeratzen da, hitz bakoitza aurreikusteko erabiliko den inguruko hitz kopurua zehazten duena. Algoritmoaren CBOW bertsioan erabiltzen den ikasketa-helburua testuinguru-hitz pare guztien egiantzaren logaritmoa izango da:

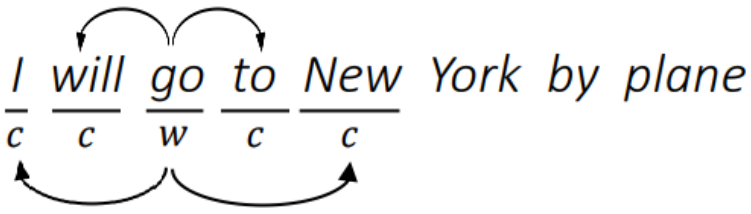
$$H = \sum_{n=c+1}^{K-c} -\log P(w_n|w_{n-c}, ..., w_{n-1}, w_{n+1}, ..., w_{n+c}),$$

hau da, testuingurua erabiliz hitza aurreikusten ikasten da. Algoritmoaren Skip-gram bertsioak, berriz, justu kontrakoa egiten du. Bertsio honetan hurrengo ikasketa-helburua minimizatzen da:

$$H = \sum_{n=1}^{K} \sum_{-c \leq j \leq c, j \neq 0} -\log P(w_{n+j}|w_n),$$

alegia, hitz bakoitza bere testuinguruko hitz guztiak aurreikusteko erabiltzen da. 3.2 irudian Skip-gram algoritmoaren funtzionamenduaren adibide bat ikusten da. Kasu honetan, testuinguru tamaina 2-koa izango da, eta *I will go to new york by plane* esaldian *go* hitza *I, will, to, new* testuinguru hitzak aurreikusteko erabiliko da. Guk lan honetan Skip-gram bertsioa erabiliko dugu embeddingak entrenatzeko.

Deskribatu dugun bezala teorikoki sarea arazorik gabe entrenatu daitekeen arren, ara-



3.2 Irudia: Skip-gram algoritmoaren funtzionamendua

zo bat aurkitu dugu erabilera praktikoan: gaur egun erabiltzen diren corpus tamainekin entrenatzea konputazionalki garestiegia da. Izan ere, ez da arraroa milaka milioika tokeneko corpusak erabiltzea; ondorioz, aurreikuspen bakoitzerako softmax osoa kalkulatzea oso garestia da. Arazo horifoz, ekiditeko algoritmoaren bi aldaera proposatu zituzten Word2Vec-en autoreek, softmax hierarkikoa eta laginketa negatiboa. Guk bigarrena erabiliko dugu lan honetan. Ikus dezagun nola aplikatu daitekeen. Lehenik score funtzioa definituko dugu, hurrengo moduan: w_i eta w_j bokabularioko bi hitz badira, eta X_j w_j hitzari dagokion one-hot bektorea bada, orduan $\text{score}(w_i, w_j) = (W_2 W_1 X_j^T)_i$ izango da, hau da, w_j hitzaren testuinguruko hitz bat w_i izatearen normalizatu gabeko (hau da, softmaxa aplikatu baino lehenagoko) probabilitatea. Skip-gram kasurako definitu dugu funtzioa; CBOW kasurako definituko bagenu, funtzioaren bigarren parametroa h testuinguru oso bat izango zen, ez w_j hitz bakarra. Orduan, hurrengo identitatea beteko da:

$$P(w_i|w_j) = \text{softmax}(\text{score}(w_i, w_j)) = \frac{\exp(\text{score}(w_i, w_j))}{\sum_{w \in Bokabulario} \exp(\text{score}(w, w_j))},$$

eta (w_i, w_j) testuinguru-hitz pare bati dagokion egiantzaren logaritmoa hurrengoa izango da:

$$\log P(w_i|w_j) = \text{score}(w_i, w_j) - \log\left(\sum_{w \in Bokabulario} \exp(\text{score}(w, w_j))\right).$$

Esan dugun bezala, softmax arrunt hau erabiltzea praktikan garestiegia da. Horren ondorioz, laginketa negatiboa erabiltzen da galera funtzioa hurbiltzeko. Laginketa negatibo teknikak galera funtzioan $\log P(w_i|w_j)$ termino bakoitza hurrengo adierazpenagatik ordezkatzen du:

$$\log \sigma(\text{score}(w_i, w_j)) + \sum_{u=1}^{k} \mathbb{E}_{w_u \sim P_n(w)} \left[\log \sigma(-\text{score}(w_u, w_j)) \right],$$

non σ ikurrak sigmoide funtzioa adierazten duen: $\sigma(x) = \frac{e^x}{e^x+1}$. Intuitiboki, galera funtzio honen esanahia hurrengoa da: benetako w_i testuinguru hitzari esleitzen zaion probabilitatea maximizatu nahi dugu, eta aldi berean $P_n(w)$ zarata banaketa batetik ausaz aukeratutako hitzei esleitzen zaien probabilitatea (bataz bestean) minimizatu nahi dugu. Hori egiten da ausaz aukeratutako hitz batek testuingurukoa izateko probabilitate oso baxua izango duelako, eta ondorioz zentzuzkoa da ausazko hitz hauen probabilitateak minimizatzea. Praktikan, formulan agertzen diren $\mathbb{E}_{w_u \sim P_n(w)} \left[\log \sigma(-\text{score}(w_u, w_j)) \right]$ esperantza matematikoak hurbiltzeko w balio bat lagintzen da $P_n(w)$ banaketatik, eta w horrekin kalkulatzen da $\log \sigma(-\text{score}(w_u, w_j))$ balioa. Zenbat eta k balioa altuagoa erabili orduan eta hurbilpen hobea lortzen da. $P_n(w)$ banaketa eta k lagin negatibo kopurua algoritmoaren parametroak dira.

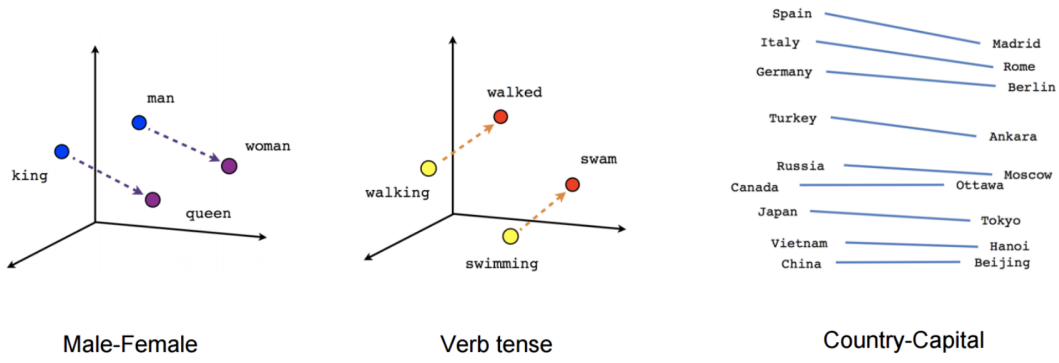
Laginketa negatiboa **NCE** edo **Noise Contrastive Estimation** galera funtzioaren hurbilpena da, eta aldi berean froga daiteke NCE softmax galeraren hurbilpena dela. Erabilera praktikoan, laginketa negatiboaren abantaila nagusia bakarrik $k+1$ score kalkulatzen direla da, softmax osoa kalkulatu behar gabe. Guk Skip-gram metodoa laginketa negatiboarekin batera erabiliko dugu gure hitz-embeddingak sortzeko.

Deskribatutako metodoa jarraituz sarea entrenatu ondoren, hitz-embeddinga W_1 matrizetik erauzten da: bokabularioko i. hitzari dagokion bektorea W_1-eko i. errenkada izango da.

Analogia Esan bezala, Word2Vec-ekin sortutako hitz-bektoreek operazio geometrikoen bidez operazio semantikoak burutzea ahalbidetzen dute. Adibide bat ikusteko, analogia operadorea honela definitzen dugu:

$$A(w_1, w_2, w_3) = \underset{w \in Bokabulario}{\arg\max} \cos(X_w, X_{w_3} + X_{w_2} - X_{w_1}),$$

non X_w w hitzari dagokion bektorea den. Hau da, operadoreak w_3 hitzaren bektoreari w_1 eta w_2 arteko desberdintasuna gehitzen dio, eta ondoren kosinu-antzekotasunaren arabera gertuen gelditzen den hitza itzultzen du. Mikolov et al.-ek frogatu zuten Word2Vec-ekin entrenatutako hitz-bektoreen bidez definitutako operadoreak analogia semantikoa egiteko balio duela. Besteak beste, ingelesez hurrengo adibideak aurkitu zituzten:



3.3 Irudia: Erlazio berdina adierazten duten bektoreak antzekoak izan ohi dira hitz-embeddingetan.

1. A(man, king, woman) = queen.

2. A(germany, berlin, france) = paris.

Beraz ikusten dugu analogia operadore honek "man is to king as woman is to queen" eta "germany is to berlin as france is to paris" motako analogiak ebatzi ditzakeela. Operadore honek funtzionatzen du hitz-embeddingaren bektore espazioan bektore batzuk erlazioak kodetzen dituztelako. Adibidez, $berlin - germany$ bektoreak herrialdetik kapitalera igarotzeko erlazioa adierazten du, eta $paris - france$ bektoreak erlazio bera adierazten du. Erlazio berdina kodetzen duten bi bektore horiek antzekoak izango dira, eta ondorioz *paris* bektorea $france + berlin - germany$ bektoretik gertu egongo da. 3.3 irudian irudikatzen da fenomeno hori.

3.1.2 Hitz-embedding notazioa

Hitz-embeddingak $\Phi : B \to \mathbb{R}^N$ mapaketa bezala definitu ditugu, non B bokabulario edo hitz multzo bat den. Askotan, bokabularioa ordenatua dagoenean, $B = \{w_1, ..., w_M\}$, embedding bat $M \times N$ tamainako X matrize baten bitartez adieraziko dugu, non X-ren i. errenkada $\Phi(w_i)$ bektorea izango den, hau da, i. hitzari dagokion bektorea. Gainera, matrizearen i. errenkada, hau da, w_i hitzaren bektorea, X_i ikurraren bidez adieraziko dugu , eta bektore honen j. elementua adierazteko X_{ij} erabiliko dugu.

Honez gain, naiz eta hitzak eta hitz-bektoreak gauza bera ez izan, askotan aurkezpena errazteko ez dugu haien artean bereiziko. Adibidez, *katu* eta *txakur* hitzei dagozkien bektoreen arteko kosinua zuzenean $\cos(txakur, katu)$ bezala adierazi dezakegu.

3.2 Hitz-embedding elebidunak

Hitz-embeddingak sortzeko proposatutako lehen metodoak kasu elebakarrean zentratzen ziren, non B bokabularioko hitz guztiak hizkuntza bakarrekoak diren. Izan ere, embeddingak sortzeko erabilitako corpus handiak elebakarrak izaten dira normalean. Hala ere, zentzuzkoa da hitz-embedding elebidunak sortzen saiatzea, non bi hizkuntzetako hitzak bektore espazio amankomun batera mapatuko diren, koherentzia mantentzen. Adibidez, euskarazko kasuan *zakur* eta *katu* hitzen bektoreak gertu egongo diren bezala, espero dezakegu ingeles-euskarazko hitz-embedding elebidun batean *zakur* eta *cat* edo *zakur* eta *dog* hitzen bektoreak elkarrengandik gertu egotea, edo $man \rightarrow king, emakume \rightarrow erregina$ motako analogia elebidunak egin ahal izatea. Orain embedding elebidunak sortzeko artearen egoeran dauden metodo batzuk azalduko ditugu.

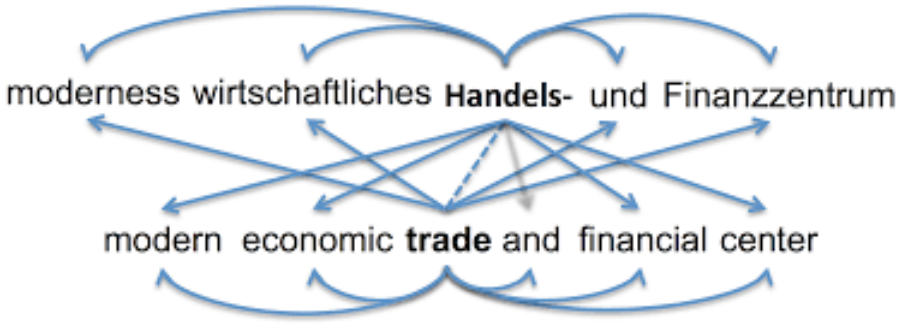
3.2.1 Aldibereko Metodoak

Metodo hauek ikasketa algoritmoa eraldatzen dute, aldi berean bi hizkuntzetako embeddingak espazio amankomunean ikasteko. Normalean, algoritmo hauek gainbegiratze elebidun maila altu bat behar dute, esaterako, esaldi mailan lerrokatutako corpus elebidun bat. Guk **BiVec** metodoa erabiliko dugu.

BiVec

Luong et al. (2015) lanean proposatutako metodo hau Skip-gramen hedapena da. Funtzionatzeko hitz-mailan lerrokatutako corpus elebidun bat behar du. Corpus elebidun lerrokatu bat honela adierazi dezakegu: Bi corpus, $C_1 = \{w_1, .., w_N\}$ eta $C_2 = \{w'_1, ..., w'_M\}$, eduki berdin-berdina baina bi hizkuntza desberdinetan izango dutenak, eta $\phi : C_1 \rightarrow C_2$ mapaketa bat C_1 corpuseko hitz bakoitza C_2 corpusean dagokion hitzera mapatuko duena.

Behin corpus lerrokatua edukita, ikasketa algoritmoa Skip-gramen berdina da, baina hitz batekin hizkuntza bereko testuingurua aurreikusteaz gain, hitz horri beste hizkuntzan dagokion hitzaren testuingurua ere aurreikusten da. Adibide bat 3.4 irudian ikusten da. Kasu honetan aleman-ingeles corpus elebidun lerrokatuan *Handels-* hitza *trade* hitzarekin lerrokatua egongo litzateke, eta *Handels-* hitza bere alemanezko corpuseko testuingurua aurreikusteko erabiltzeaz gain, *trade* hitzaren ingelesezko testuingurua aurreikusteko ere erabiliko da, eta alderantziz. Honela, Skip-gram arruntak hizkuntza berean semantikoki antzekoak diren hitzei bektore antzekoak esleitzen dizkien bezala, BiVec algoritmoak



3.4 Irudia: BiVec algoritmoa irudikatua. Kreditua Luong et al. (2015).

esanahi antzekoak dituzten hitzei (naiz eta hizkuntza desberdinetakoak izan) antzeko bektoreak esleituko dizkie. Hau da, BiVec algoritmoak bi hizkuntzetako hitzak zuzenean bektore espazio amankomun batera mapatuko ditu.

Metodo honek oso emaitza onak lortzen ditu, baina eskatzen duen gainbegiratze maila oso altua da. Dagoeneko zaila izaten da corpus elebakar handiak lortzea embeddingak entrenatzeko, eta eskuragarri dauden corpus elebidun lerrokatuak corpus elebakarrak baino askoz txikiagoak izaten dira, batez ere hizkuntza pare arraroentzat. Adibidez, ingelesez eta finlandiarrez bilioi bat token baino gehiagoko corpus elebakarrak existitzen dira [2] [3], baina ingeles-finlandiar parerako ParaCrawl corpusak ingeles aldean 54 milioi token baino ez ditu [4].

3.2.2 Mapaketa-metodoak

Mapaketa-metodoek hurbilpen guztiz desberdina erabiltzen dute: lehenik corpus eta hitz-embedding algoritmo elebakarrak (Word2Vec, adibidez) erabiliz hizkuntza bakoitzeko hitz-embeddingak independenteki sortzen dira, eta bigarrenik transformazio linealak erabiliz bi bektore espazioak espazio amankomun batera mapatzen dira. Zehazki, demagun $B = B_1 \bigcup B_2$ bokabulario elebidun bat dugula, B_1 eta B_2 bokabulario elebakarrek osatua. B_1 bokabularioaren hizkuntzari jatorri hizkuntza eta B_2-renari helburu hizkuntza deitzen zaio. Orduan, B-ren hitz-embeddinga lortzeko, lehenik $\Phi_1 : B_1 \to \mathbb{R}^N$ eta $\Phi_2 : B_2 \to \mathbb{R}^N$ embedding elebakarrak ikasten dira, eta $\Phi : B \to \mathbb{R}^N$ embeddinga honela definitzen da:

[2] https://nlp.stanford.edu/projects/glove/
[3] https://www.sketchengine.eu/fitenten-finnish-corpus/
[4] https://paracrawl.eu/releases.html

$$\Phi(w) = \begin{cases} T_1\Phi_1(w) & \text{baldin } w \in B_1 \\ T_2\Phi_2(w) & \text{baldin } w \in B_2 \end{cases}$$

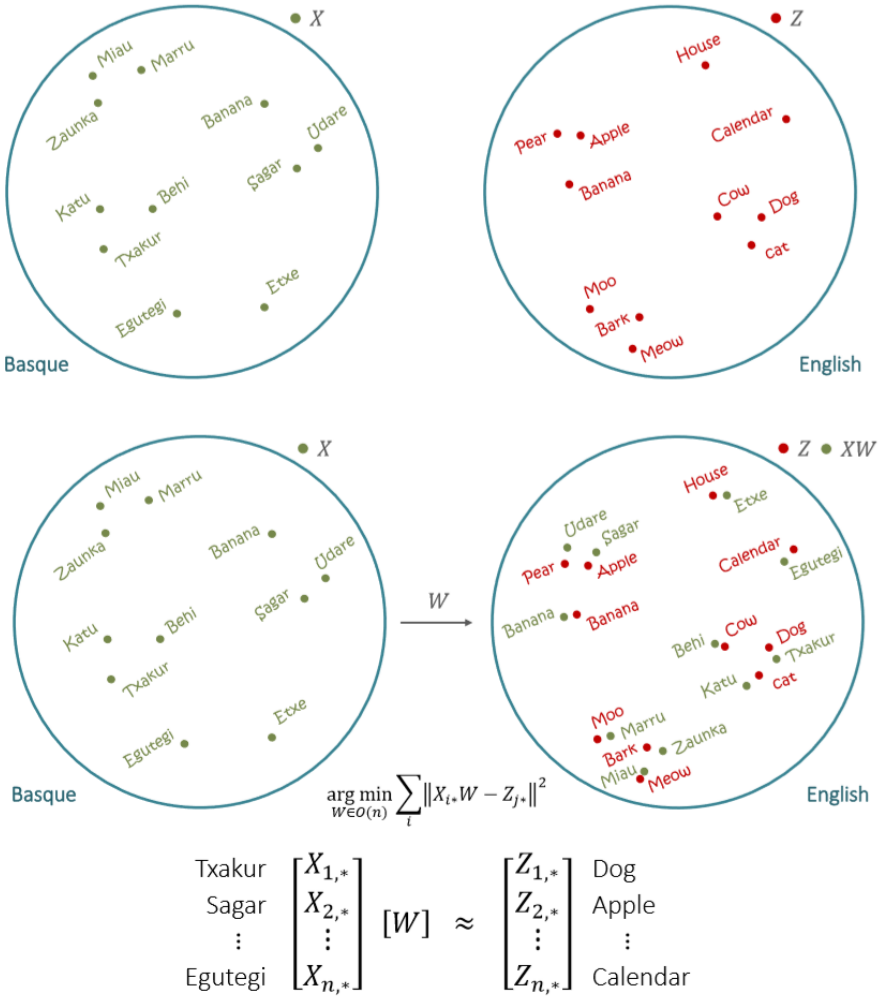
f non $T_1, T_2 : \mathbb{R}^N \to \mathbb{R}^N$ mapaketa linealak izango diren. Matrize notazioak erabiltzen, embedding matrizeak X eta Y badira, orduan mapaketa linealak W_1 eta W_2 matrizeen bidez adierazi ditzakegu, embedding mapatuak XW_1, YW_2 izango dira, eta bi matrize hauen errenkadetan egongo dira embedding elebidunaren hitz-bektoreak. Beraz metodo honen gakoa W_1 eta W_2 matrize egokiak ikastean datza. Horretarako, ezinbestekoa da $\Phi_1(B_1)$ eta $\Phi_2(B_2)$-ren egitura geometrikoa uztartzea. Izan ere, pentsatzekoa da hizkuntza desberdinetan, naiz eta hitz desberdinak erabili kontzeptuak deskribatzeko, kontzeptu horien erabilera antzekoa egiten dela, eta ondorioz Word2Vec motako algoritmo batekin sortutako hizkuntza desberdinetako hitz-embeddingek egitura geometriko antzekoa edukiko dutela. Azken hori isomorfia hipotesia da, independenteki ikasitako hizkuntza desberdinetako hitz-embeddingak isomorfoak izatetik hurbil egongo direla diona (Miceli Barone, 2016).

Hipotesi hori onartzen badugu, pentsatu dezakegu azken finean independenteki ikasitako embedding desberdinak berdinak direla (isomorfoak), baina ez daudela orientazio berdinean (transformazio lineal batek desberdintzen ditu). Beraz, posible izango litzateke erreferentzia puntu gutxi batzuk erabiliz bi espazioak berriro lerrokatzea transformazio lineal baten bidez. Ideia hori da Mikolov et al.-ek erabili zutena. Hurrengo metodoa proposatu zuten: Demagun L_1 eta L_2 bi hizkuntzen hitz-embeddingak sortu ditugula, X eta Y. Gainera, $H = \{(X_{i_1}, Y_{j_1}), ..., (X_{i_m}, Y_{j_m}\}$ hiztegi elebidun bat daukagula suposatzen da. Hiztegi honen elementuak (X_i, Y_j) motako pareak dira, L_1 hizkuntzako hitz baten bektorea eta hitz horren L_2-ko itzulpenari dagokion bektorea gordetzen dutenak. Orduan, W_2 transformazioa identitatea izango da, eta W_1 transformazioa hurrengo ikasketa-helburua minimizatuz aurkitzen da:

$$W_1 = \underset{W_1 lineala}{\arg\min} \sum_{(X_i, Y_j) \in H} ||X_i W_1 - X_j||^2,$$

hau da, hiztegiko hitz pareen arteko distantzien karratuen batura minimzatzen duen transformazio lineala erabiltzen da. Espazioen egitura antzekoa denez, hiztegiko erreferentzia puntuen arteko distantzia minimizatuz lortzen da hiztegitik kanpo dauden puntuak ere ondo lerrokatzea. 3.5 irudian prozesu hau irudikatzen da.

Ikerkuntza lerro hau oso aktiboa izan da azken urteetan, eta azaldu dugun lehen metodoa-



3.5 Irudia: Hiztegi bidezko mapaketa gainbegiratuaren irudikapena. Kreditua Mikel Artetxe.

ren hobekuntza asko egon dira. Bereziki aipagarria da lortu izan dela mapaketa ikasteko hiztegi baten beharra ekiditea, eta badaudela metodoak modu guztiz gainbegiratuan ikasten dituztenak mapaketak (Zhang et al., 2017; Conneau et al., 2018; Artetxe et al., 2018b).

Lan honetan artearen egoeran dauden bi mapaketa-metodo erabili ditugu: **Vecmap** eta **RCSLS**.

3.2.3 Vecmap

Metodo honen hainbat bertsio existitzen dira, batzuk gainbegiratuak eta beste batzuk ez-gainbegiratuak. Lehenik aldaera ez-gainbegiratua azalduko dugu.

Vecmap ez-gainbegiratua

Vecmapen aldaera hau Artetxe et al. (2018b) lanean proposatutakoa da. Algoritmo honek honako urratsak jarraitzen ditu embedding mapaketa ikasteko:

- Embedding normalizazioa
- Hasieraketa heuristikoa
- Bilaketa iteratibo estokastikoa
- Re-weighting simetrikoa

Atal honetan, jatorri eta helburu hizkuntzen embeddingak X eta Y matrizeen bidez adieraziko ditugu. Bi bokabularioak M tamainakoa badira, eta hitz-bektoreak N dimentsiokoak, orduan matrize horiek $M \times N$ dimentsiokoak izango dira, eta mapaketa linealak W_1 eta W_2 $N \times N$ matrizeen bidez adieraziko ditugu. Helburua, beraz, XW_1 eta YW_2 espazio berdinera mapatzen dituzten W_1 eta W_2 matrizeak ikastea izango da. Embeddingeko i. bektorea adierazteko X_i eta Y_i erabiliko dugu.

Embedding normalizazio pausoan, jatorri eta helburu hizkuntzako embeddingak luzera-normalizatu egiten dira (hau da, bektore bakoitza bere moduloarengatik zatitzen da), ondoren zentratu egiten dira (bektore bakoitzari bektore guztien bataz bestekoa kentzen zaio), eta azkenik berriro luzera-normalizatzen dira. Normalizazioaren ondorioz, bektoreen arteko produktu eskalarrak kosinu-antzekotasuna emango digu.

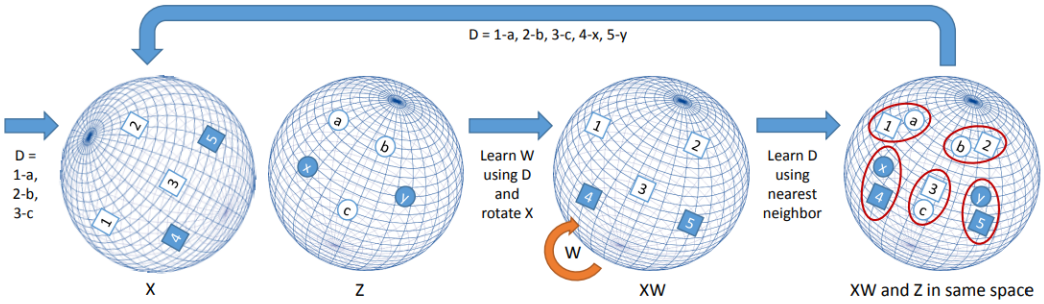
Jarraian, hasieraketa heuristiko pausoaren helburua modu ez-gainbegiratuan hasierako hiztegi elebidun bat ikastea da, kalitate txikikoa izango dena, baina bilaketa hasieratzeko nahikoa. Horretarako, lehen aipatu dugun isomorfismo hipotesiaz baliatzen da metodoa. Ikusten dugu $M_X = XX^T$ eta $M_Y = YY^T$ matrizeak bi embedding espazioen antzekotasun-matrizeak direla; izan ere, XX^T-ko i lerroko eta j zutabeko elementua $X_i X_j^T = \cos(X_i, X_j)$ izango da. Orduan, suposatzen badugu X eta Y embeddingak gutxi gora behera isomorfoak direla espazio bakoitzeko puntuen arteko kosinu-antzekotasun erlatiboak berdinak izango dira, eta ondorioz M_X eta M_Y matrizeen arteko alde bakarra lerro eta zutabeen permutazio bat izango da. Praktikan, isomorfia hipotesia ez da guztiz betetzen, baina espazioen antzekotasuna nahikoa izango da. Hala ere, permutazio egokia aurkitzeko M_X eta M_Y matrizeen zutabe eta lerroen permutazio guztiak probatzea konputazionalki garestiegia da, eta ondorioz hurrengo hurbilpena erabiltzen da: M_X eta M_Y matrizeetako lerro bakoitzeko balioak ordenatzen dira, eta horrela $sorted(M_X)$ eta $sorted(M_Y)$ matrizeak lortzen dira. Teorian, isomorfia hipotesia guztiz beteko balitz, hitz baliokideek bektore berdin-berdina edukiko lukete $sorted(M_X)$ eta $sorted(M_Y)$ matrizeetan, eta ondorioz pentsatu dezakegu $sorted(M_Y)$-ko errenkadei gertukoen-auzokide atzipena (*nearest-neighbor retrieval* ingelesez) aplikatuz dagozkien $sorted(M_X)$-ko errenkada aurkitzea posible izango dela [5]. Praktikan, autoreek aipatu zuten emaitza hobeak lortzen zirela $sorted(\sqrt{M_X})$ eta $sorted(\sqrt{M_Y})$ matrizeak erabiltzen. Beraz, pauso honetan bi azken matrize horiek kalkulatu eta embedding normalizazio pausoan azaldutako moduan normalizatzen dira, X' eta Y' matrizeak lortzeko. Bi matrize hauen bidez gertukoen-auzokide atzipena erabiliz hiztegi bat kalkulatuko da, bilaketa iteratibo estokastiko pausoa martxan jartzeko erabiliko dena.

Bilaketa pausoan, iteratiboki mapaketa gero eta hobeagoak induzituko dira heuristikoki lortutako hasierako soluzioa hobetzeko. Honetarako, suposatuz uneko iterazioan D hiztegi elebidun bat dugula (hiztegia D matrize baten bidez adierazten da, non $D_{ij} = 1$ izango den X-ko i. hitzaren itzulpena Y-ko j. hitza bada, eta $D_{ij} = 0$ bestela), hurrengoa egiten da:

- Lehenik hurrengo hiztegiko hitz pareen arteko kosinu-antzekotasunen batura minimizatzen duen W_1 eta W_2 matrize ortogonalak kalkulatzen dira, hau da:

$$\underset{W_1\ W_2\ ortogonalak}{\arg\max} \sum_i \sum_j D_{ij}((X_i W_1) \cdot (Y_j W_2)).$$

[5] Gertukoen-auzokide atzipena egitean x bektore bati $Y = \{y_1, ..., y_n\}$ multzo batean dagokion bektorea esleitzeko gertukoena aukeratzen da, hau da, $y \in Y\ \forall y' \in Y d(x,y) \leq d(x,y')$ betetzen duena. Beraz atzipen metodo hau erabilitako distantzia neurriaren araberakoa izango da.



3.6 Irudia: Vecmap ez-gainbegiratuaren pauso iteratiboa irudikatua. Kreditua Artetxe et al. (2017).

- Bigarrenik, gertukoen-auzokide eta kosinu-antzekotasun distantzia erabiliz hiztegi berri bat induzitzen da, hau da, $D_{ij} = 1$ izango da $j = \arg\max_k(X_i W_1) \cdot (Y_k W_2)$ betetzen bada, eta $D_{ij} = 0$ bestela.

Autoreek frogatu zuten pauso hau iteratiboki egikarituz algoritmoak inplizituki galera funtzio bat minimizatzen duela, eta minimo lokal batera konbergituko duela beti. Gainera, praktikan algoritmoak hobeto funtzionatzeko artikuloan hainbat hobekuntza azaltzen dira:

- Hiztegi indukzio estokastikoa: D_{ij} matrizeko elementu batzuk ausaz zerora ezartzen dira, bilaketan esplorazioa sustatzeko.
- Frekuentzian oinarritutako bokabulario-kimaketa. Bakarrik maiztasun handieneko k hitzentzat induzitzen da D_{ij} hiztegia gertukoen-auzokide atzipen bidez, eta hiztegi hori erabiltzen da hurrengo mapaketa aurkitzeko.
- Bi noranzkoko hiztegi indukzioa: hiztegia bakarrik jatorri hizkuntzako hitz bakoitzari helburu hizkuntzako gertukoen auzokidea esleituz induzitu ordez, indukzioa kontrako norabidean ere egiten da. Hau da, $D_{ij} = 1$ izango da ere $i = \arg\max_k(X_k W_1) \cdot (Y_j W_2)$ betetzen bada.
- CSLS atzipena: hiztegi indukzio pausoan gertukoen-auzokide atzipena egitean geroago azalduko den CSLS distantzia erabiltzen da.

Bilaketa prozesua martxan jartzeko hasierako hiztegia lehen azaldutako heuristikotik lortutako X' eta Y' matrizeen bidez kalkulatzen da; ondoren, matrize horiek baztertu egiten dira eta X, Y matrizeak erabiltzen dira hortik aurrera.

3.6 irudian Vecmap ez-gainbegiratuaren pauso iteratiboa irudikatzen da. Ikusten da uneko D hiztegia erabiltzen dela mapaketa bat ikasteko, eta ondoren mapatutako espaziotik hiztegi berri bat erauzten dela.

Azkenik, re-weighting pausoa egiten da. Pauso honetan $X^T DY$ matrizearen (non D azken induzitutako hiztegia den) SVD dekonposaketa kalkulatzen da, $USV^T = X^T DY$, jarrain $W_x = US^{\frac{1}{2}}$ eta $W_Y = VS^{\frac{1}{2}}$ matrizeak kalkulatzen dira, eta azkenik matrize horiek pauso iteratiboan lortutako W_1, W_2 matrizeei biderkatzen zaizkie azken matrizeak lortzeko. Autoreek azaltzen dute pauso honek mapatutako embeddingen artean bat datozen dimentsioen garrantzia areagotzen duela, eta honela mapaketa hobea lortzen dela.

Vecmap gainbegiratua

Algoritmo gainbegiratu honek, Artetxe et al.-ek proposatua, aldaera ez-gainbegiratuak jarraitzen dituen pauso berdin-berdinak jarraitzen ditu, baina hasieraketa heuristikoa eta pauso iteratiboa egin gabe. Hau da, bertsio honetan embeddingak normalizatzen dira, ondoren emandako D hiztegia erabiliz W_1 eta W_2 matrizeak aurkitzen dira, eta azkenik re-weighting egiten da. Desberdintasun bakarra hiztegia heuristiko baten bidez sortu ordez kanpotik jasotzen dela da, eta ez dela mapaketa-hiztegi indukzio pausoa iteratiboki errepikatzen.

Vecmap semi-gainbegiratua

Vecmapen bertsio hau ez-gainbegiratua bezalakoa da, baina hasierako hiztegia heuristikoaren bidez induzitu ordez hazi-hiztegi txiki bat ematen zaio algoritmoari, hasierako soluzioa eraiki ahal izateko. Ondoren gainontzeko guztia berdin egiten da. Bertsio hau erabilgarria da hiztegi handi bat ez daukagunean, baina metodo ez-gainbegiratua ez denean gai hasieraketa heuristikotik abiatuta optimo lokal ona aurkitzeko (normalean heuristikoaren bidez lortutako hasierako soluzioa txarregia izan delako).

Halaber, posible da metodo hau modu ia ez-gainbegiratuan erabiltzea, hazi-hiztegia automatikoki bi hizkuntzetan berdin idazten diren hitzetatik erauziz. Normalean hizkuntza desberdinetan esanahi antzekoa duten eta berdin idazten diren hitzak egoten dira, eta posible da hitz horiek erabiltzea hasierako soluzioa lortzeko. Automatikoki sortutako hiztegi hau nahikoa izaten da pauso iteratiboan soluzio on batera konbergitu ahal izateko (Artetxe et al., 2017).

3.2.4 RCSLS

Metodo hau, Joulin et al. (2018) lanean proposatua, atalaren hasieran deskribatu dugun antzera gainbegiratua da, galera funtzio baten minimizazioan oinarritua, baina W_1 mapaketa aurkitzeko ikasketa-helburuan distantzia euklidearra ordez CSLS izeneko neurria erabiltzen da:

$$W_1 = \underset{W_1 \; lineala}{\arg\max} \sum_{(X_i,Y_j)\in H} \text{CSLS}(X_i W_1, Y_j),$$

CSLS neurria 3.3.1 azpiatalean azalduko dugu.

3.3 Embedding elebidunen ebaluazioa

Literaturan hitz-embedding elebidunak ebaluatzeko **HEI** (**Hiztegi Elebidun Indukzioa** , edo *Bilingual Lexicon Induction* ingelesez) neurria oso erabilia da.

HEI ebaluazioa kalkulatzeko, lehenik embedding elebiduna erabiltzen da hizkuntzen arteko hiztegi bat induzitzeko, eta bigarrenik sortutako hiztegia urre patroi hiztegi batekin alderatzen da. Zehazki, demagun jatorri hizkuntzaren bokabularioa B_1 dela eta helburu hizkuntzarena B_2 dela. Guk erabiliko ditugun hiztegiak $\phi : A \subset B_1 \rightarrow \mathcal{P}(B_2)$ [6] funtzio baten bidez adieraz daitezke: funtzio honek jatorri hizkuntzako A multzoko hitz bakoitzari bere itzulpena esleitzen dio, non itzulpena B_2-ko azpimultzoa izango den, polisemia kontuan hartzeko (adibidez, euskara-ingeles kasuan *heldu* hitzaren itzulpena $\phi(heldu) = \{arrive, hold, ripen\}$ izan daiteke).

Ondoren, $\Phi : B_1 \cup B_2 \rightarrow \mathbb{R}^N$ embedding elebiduna ebaluatzeko, embedding elebiduna erabiliz A multzoko hitz bakoitzaren itzulpena kalkulatzen dugu, zehaztasuna neurtzen dugu. Zehazki, $T : B_1 \rightarrow B_2$ operadoreak Φ embedding elebidunaren bidez egindako itzulpena adierazten badu, orduan zehaztasuna horrela kalkulatzen dugu [7]:

$P@1 = \frac{|\{w\in A : T(w)\in\phi(w)\}|}{|A|}$,

hau da, hiztegiko hitzen itzulpenak egitean zein proportzio asmatu den izango da ebaluazio neurria.

[6] Hemen $\mathcal{P}$ ikurrak parteen multzoa adierazten du, hau da, S multzo bat bada $\mathcal{P}(S)$ S-ren azpimultzo guztiek osatutako multzoa izango da.

[7] Zehaztasuna $P@1$ ikurraren bidez adierazten dugu, ingelesezko *precision at one* terminoan oinarrituta.

Hainbat aplikazio praktiko dituen ataza izateaz gain, HEI ebaluazioa embeddingen kalitatearen neurri adierazgarria da, hitzen errepresentazio elebidun on batek esanahi antzekoak dituzten hitzak elkarrengandik gertu kokatu behar dituelako.

Ebaluazio metodo hau guztiz zehazteko itzulpenak atzitzeko metodoa erabaki behar da, hau da, T operadorea. Horretarako, $\mathbb{R}^N$ espazioan $d : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$ distantzia neurri bat definitzen da, eta T operadorea honela definitzen da:

$$T(w) := \arg\min_{w' \in B_2} d(w, w'),$$

hau da, jatorri hizkuntzako w hitz baten itzulpena helburu hizkuntzan gertuen dagoen w' hitza izango da, d distantzia neurriaren arabera. Literaturan neurri erabiliena $d(w, w') = \cos(w, w')$ kosinu-antzekotasuna da (naiz eta matematikoki distantzia metrika ez izan). Orain atzipena egiteko erabili daitekeen beste neurri bat ikusiko dugu, CSLS izenekoa.

3.3.1 CSLS

Geroago ikusiko dugun bezala, hitz-embedding elebidunetan hubness [8] izeneko fenomenoa agertzen da. Fenomeno horrek espazioko hitz gutxi batzuk, hub deritzogunak, beste hitz askoren gertukoen auzokideak (eta ondorioz haien itzulpenak HEI egitean) izatea eragiten du. Gainera, hubnessak eragin negatiboa du HEIn. Arazo horri aurre egiteko, Conneau et al.-ek hubnessa kontuan hartzen duen distantzia neurri egokitu bat erabiltzea proposatu zuten, CSLS deitu zutena. X eta Y bi hizkuntzetako embedding matrizeak badira, eta $x = X_i, y = Y_j$ bi hitz-bektore badira, CSLS antzekotasuna honela definitzen da:

$$\mathrm{CSLS}(x, y) = 2\cos(x, y) - \frac{1}{k} \sum_{y' \in \mathcal{N}_Y(x)} \cos(x, y') - \frac{1}{k} \sum_{x' \in \mathcal{N}_X(y)} \cos(x', y).$$

Aurreko formulan $\mathcal{N}_X(y)$ multzoa X multzoan y bektoretik gertuen dauden k bektoreen multzoa da, kosinu-antzekotasuna erabilita kalkulatua (hau da, y bektorearen X-ko k-gertukoen-auzokide multzoa), eta $\mathcal{N}_Y(x)$ modu berean definitzen da.

Definizioak hiru termino ditu: lehena x eta y arteko kosinu-antzekotasun arruntaren bikoitza da. Bigarren terminoak x eta x-ren k elementu gertukoenen (non gertukoen elementu hauek kontrako hizkuntzaren espazioan, hau da, Y multzoan, hartzen diren) arteko bataz

[8] *Hubness* terminoa zabaldua dago ingelesezko literaturan, eta ez dugu euskaratzeko modu argirik aurkitu, beraz itzuli gabe mailegutzat hartu dugu testuan zehar.

besteko kosinu-antzekotasuna adierazten du, eta azken terminoa bigarrenaren berdina da, baina x ordez y erabiliz kalkulatua.

Definizioari bigarren eta hirugarren terminoak gehitzearen motibazioa hurrengoa da: y espazioko hub indartsua bada, litekeena da kontrako hizkuntzan hainbat bektore oso gertu edukitzea (hitz askoren itzulpena delako), eta ondorioz $\frac{1}{k}\sum_{x' \in \mathcal{N}_X(y)} \cos(x', y)$ terminoa altuagoa izango da. Hitza ez baldin bada hub bat, berriz, litekeena da inguruko hitzak hain gertu ez egotea , eta ondorioz termino hau ez da hain altua izango. Horrela, CSLS-ren definizioko bigarren eta hirugarren terminoek espazioko hubak penalizatzen dituzte, eta espero dezakegu HEI egitean itzulpen bezala benetako itzulpena aukeratzea hub hitz baten ordez.

HEI egiteko kosinu-antzekotasun bidezko atzipena erabiltzen denean NN atzipena erabili dela esaten dugu, eta CSLS neurria erabiltzean CSLS atzipena erabili dela esaten dugu. Conneau et al.-ek frogatu zuten CSLS atzipena erabiltzean HEI kalitatea nabari hobetzen zela.

3.4 Hubness

Fenomeno ezaguna da dimentsio altuko espazioetan puntu multzoak lagintzean puntu gutxi batzuk beste puntu askoren gertukoen auzokide izateko joera dutela, eta horri **hubness** fenomenoa deritzogu. Zehazki, demagun $A \subset \mathbb{R}^N$ puntu multzoa dugula, eta puntuen arteko d distantzia neurria dugula. $x \in A$ bakoitzeko $NN(x)$ honela definitzen dugu:

$$NN(x) = |\{y \in A : d(x,y) = \min_{y' \in A} d(y,y')\}|.$$

Hau da, $NN(x)$ balioak adierazten du zenbat puntu dauden A-n zeinen gertukoen auzokidea x den. Orduan, $NN(x)$ balio altuko x puntuei **hub** deritzegu, eta espazioan hub asko daudenean hubness handia dagoela esaten dugu.

Frogatua izan da, N dimentsioa hazten doan heinean, baldintza nahiko orokorretan (adibidez, puntuak banaketa normal batetik lagintzean, edo hainbat dataset enpirikoetan) puntu multzoen hubnessa hazi egiten dela (Radovanović et al., 2010a). Ikusiko dugun bezala, hubness fenomenoak eragin kaltegarria du hitz-embedding elebidunetan.

3.4.1 Embedding elebidunetan

Testuinguru elebidunean modu antzekoan neurtu dezakegu hubnessa. X eta Y badira L_1 eta L_2 hizkuntzetako bektoreen multzoak, orduan $y \in Y$ bada honela definitzen dugu $NN(y)$:

$$NN(y) = |\{x \in X : d(x,y) = \min_{y' \in Y} d(x,y')\}|,$$

eta $x \in X$ bada $NN(x)$ honela definitzen dugu:

$$NN(x) = |\{y \in X : d(x,y) = \min_{x' \in A} d(y,x')\}|,$$

hau da, $NN(x)$ neurriak adierazten du HEI egitean L_2 hizkuntzako zenbat hitzen itzulpena izango den x, eta $NN(y)$-k adierazten du L_1 hizkuntzako zenbat hitzen itzulpena izango den y. Kasu elebakarrean bezala, NN balio altua duten hitzak hubak direla esaten dugu.

Argi dago horrela neurtutako hubnessa altua denean, HEI ebaluazioan eragin kaltegarria izango duela. Adibidez, euskara-ingeles parean y ingelesko *dog* hitzaren bektorea bada, eta $NN(y) = 100$ badugu, horrek esan nahi du *dog* euskarazko 100 hitzen itzulpena izan dela. Naiz eta hizkuntzak polisemikoak izan, argi dago hitz bakar bat 100 hitzen itzulpena izatea ez dela zuzena, eta ondorioz euskarazko hainbat hitz oker itzuliko dira.

Fenomeno ezaguna da mapaketa bidez sortutako embedding elebidunek hubness handia sufritzen dutela. Are gehiago, askotan hub indartsuenak ez dira zentzuzkoak izaten (hau da, ez dira naturalki hitz askoren itzulpen izan daitezkeen hitzak izaten).

Hubnessa eta CSLS

HEI egitean gertatzen den bezala, hitzen hubnessa neurtzeko definitu dugun NN metrika d distantzia funtzio baten araberakoa da. Distantziak neurtzeko kosinu-antzekotasun estandarra erabiltzen denean, esaten dugu NN atzipen bidez kalkulatu dela hubnessa, eta CSLS erabili denean esaten dugu CSLS atzipena erabili dela.

3.5 Isomorfia

Esan dugun bezala, embedding elebidunak sortzeko mapaketa-metodoen eraginkortasuna isomorfia hipotesiaren mendekoa da. Izan ere, independenteki ikasitako embeddingek

egitura guztiz desberdinak badituzte, ezinezkoa izango da modu ez-gainbegiratuan bi espazioak lerrokatzea mapaketa lineal baten bidez.

Hala ere, berriki autore batzuk isomorfia hipotesi hau zalantzan jarri dute. Søgaard et al. (2018) laenan autoreek frogatu zuten independenteki ikasitako hizkuntza desberdinen embeddingak ez direla isomorfoak, eta hizkuntza pare batzuen kasuan isomorfoak izatetik oso urrun daudela.

Lan honetako esperimentuetan oso erabilgarria izango da isomorfia metrika bat, hau da, bi embeddingen egituren arteko antzekotasunaren neurri bat izatea. Horretarako Søgaard et al.-ek proposatutako isomorfia neurria erabiltzen dugu. Neurri horrek embedding bakoitzeko gertukoen-auzokide grafoen matrize laplaziarren autobalioen arteko antzekotasuna neurtzen du. Zehazki definitzeko, demagun X eta Y bi embeddingen bektore multzoak direla. Orduan, G_1 grafo ez-zuzendua definitzen dugu hurrengo moduan: erpin multzoa X izango da, eta $x_1 \neq x_2$ badira, $\{x_1, x_2\}$ ertza E ertz multzoan egongo da hurrengo baldintza betetzen badu:

$$\{x_1, x_2\} \in E \iff d(x_1, x_2) = \min_{x \in X} d(x_1, x) \vee d(x_1, x_2) = \min_{x \in X} d(x, x_2).$$

Hau da, x_1 eta x_2 elementuen artean ertz bat egongo da bietako bat bestearen gertukoen auzokidea baldin bada. Modu berean G_2 grafoa definitzen da Y espaziorako. Ondoren, G_1 eta G_2 grafoen L_1 eta L_2 matrize laplaziarrak kalkulatzen ditugu. Jarraian, hurrengo propietatea betetzen duen k_1 txikiena aukeratzen dugu: L_1 matrizearen lehen k_1 autobalio handienen batura gutxienez matrizearen autobalio guztien baturaren %90 izan behar da. Modu berean k_2 balioa kalkulatzen dugu L_2 matrizearentzat. Azkenik, $k = \min(k_1, k_2)$ jartzen dugu, eta X eta Y arteko autobalio antzekotasuna honela definitzen dugu:

$$\Delta = \sum_{i=1}^{k} (\lambda_{1i} - \lambda_{2i})^2,$$

non $\{\lambda_{1i}\}$ eta $\{\lambda_{2i}\}$ L_1 eta L_2 matrizeen autobalioak diren, handienetik txikienera ordenatuak.

Autoreek frogatu zuten autobalio antzekotasun neurriak eta embedding elebidunen kalitateak (HEI bidez neurtua) korrelazio indartsua dutela.

4. KAPITULUA

Hubness aztertzen

Aurrekarien kapituluan ikusi den bezala, fenomeno ezaguna da hitz-embedding elebidunek hubnessa sufritzen dutela. Gainera, ikusi da hubnessak eragin kaltegarria duela embeddingen kalitatean, eta ondorioz interesgarria da fenomeno hau ondo ulertzea, posible bada konpondu edo arintzeko helburuarekin. Hubnessa ulertzeko egindako esperimentuak eta erauzitako ondorioak azalduko ditugu kapitulu honetan.

Hasiera batean, hitz-embedding elebidunetan agertzen den hubnessaren bost iturri edo eragile posible pentsatu genituen:

- Domeinu desberdintasunak eragiten du hubnessa. Esaterako, medikuntzako corpus eta corpus orokor batzuen bidez entrenatutako bi embedding elebakar espazio amankomun batera mapatzean hubness altua agertuko da. Hipotesi honek mapaketa bidez sortutako embeddingei egiten die erreferentzia soilik, aldibereko ikasketaren kasuan ez delako posible domeinu desberdintasunak egotea (corpus elebidun lerrokatua erabiltzen delako).
- Hizkuntzen desberdintasunak, ikuspuntu linguistiko batetik, eragiten du hubnessa. Adibidez, frantsesa eta italiera nahiko antzekoak dira linguistikoki, aldiz, finlandiera desberdinagoa da (hizkuntza ez-indoeuropear aglutinatibo bat baita), eta ondorioz posible da FR-IT hizkuntzetako hitz-embedding elebidunetan hubness baxuagoa agertzea FR-FI edo IT-FI kasuetan baino.
- Mapaketa-metodoen araberakoa da hubnessa. Autore batzuk argudiatu dute transformazio linealak lortzeko erabiltzen diren ikasketa-helburu batzuk hubness altua

sortzea eragin dezaketela (Shigeto et al., 2015). Ondorioz, pentsatu dezakegu mapaketa algoritmo batzuk hubness altuagoko embeddingak sortuko dituztela. Are gehiago, mapatze prozesuak berak hubnessa igotzen badu, posible da mapaketa-metodo iteratiboek hubness altuagoko soluzioak aurkitzea, behin eta berriz mapatzen baidute.

- Hubnessa embedding elebidun on baten ezaugarri bat da. Intuitiboa ez den arren, posible da hubness altua hitz-embedding elebidun on baten berezko ezaugarri bat izatea, eta kasu horretan bai mapaketa bidez eta bai aldibereko ikasketa bidez sortutako hitz-embedding elebidunetan hubness altua agertuko litzateke.

Faktore hauen eta hubnessaren arteko erlazioa aztertzeko egindako esperimentuak azalduko ditugu hurrengo ataletan.

4.1 Esperimentuaren diseinua

Sinpleki, esperimentua hurrengoa izango da: hainbat hizkuntza/domeinu pare eta mapaketa-metodo desberdin erabiliz hitz-embedding elebidunak sortuko ditugu, eta corpus/metodo konbinazio bakoitzarekin sortutako embeddingetan agertzen den hubnessa aztertuko dugu. Era berean, embedding bakoitzaren kalitatea ere ebaluatuko dugu, hubnessarkin duen harremana aztertu ahal izateko.

Esperimentua egikaritu ahal izateko, hurrengo faktoreak zehaztu behar ditugu:

1. Metodoak.
2. Hizkuntza pareak eta entrenamendu corpusak eta hiztegiak.
3. Hubnessa neurtzeko metrikak.
4. Ebaluazio metrikak.

4.1.1 Metodoak

Erabiliko ditugun metodoak, 3 kapituluan azalduak, honako hauek dira:

- **BiVec** aldibereko metodoa, hurrengo parametroekin: 1e-5eko sub-sampling, 10eko laginketa negatiboa, 300eko bektore dimentsioa, eta 5 entrenamendu iterazio. Emaitza tauletan eta eztabaidan BiVec deritzogu metodo honi.

- **Vecmap** gainbegiratua, lehenetsitako parametroekin. Emaitza tauletan eta eztabaidan Sup izenarekin adierazten dugu metodo hau.

- **Vecmap** ez-gainbegiratua, lehenetsitako parametroekin. Emaitza tauletan eta eztabaidan Unsup izenarekin adierazten dugu metodo hau.

- **Vecmap** semi-gainbegiratua, lehenetsitako parametroekin. Kasu honetan, bertsio semi-gainbegiratuaren bi aldaera erabiliko ditugu: lehena, hitz kognatuez (bi hizkuntzetan berdin idazten diren hitzak) osatutako hiztegiarekin hasieratua, eta bigarrena, hitz bakarreko ausazko hiztegi batekin (adibidez *mahai-dog* ausaz aukeratutako pare bakar batez osatutako hiztegi batekin) hasieratua. Hitz kognatuekin hasieratutako bertsioa ia ez-gainbegiratua da, hasierako hiztegia automatikoki bokabularioetatik erauzten delako, eta aldaera ez-gainbegiratuaren ordezkapen bezala erabili dezakegu, azken horrek ez duenean soluzio on bat ematen (esaterako, hasieraketa heuristikoak ez duelako ondo funtzionatu). Ausazko hiztegiarekin hasieratzean, algoritmo semi-gainbegiratuko bilaketa pausoak optimo lokal txar batera konbergituko du normalean, eta algoritmo hau gehitzearen arrazoia ea optimo lokal txar batera konbergitzean hubness desberdina agertzen den ikustea da. Kognatuekin hasieratutako metodo ez-gainbegiratua Ident izenarekin adierazten dugu emaitza tauletan eta eztabaidan, eta ausaz hasieratutakoa Semisup izenarekin.

- **RCSLS** mapaketa-metodoa, 25eko learning rate eta 10 ikasketa iteraziorekin. RCSLS izenarekin adierazten dugu metodo hau emaitza tauletan.

Metodo guztiak autoreek eskuragarri jarritako inplementazioekin egikaritu ditugu [1] [2] [3]. Gainera, mapaketa-metodoen kasuan erabiliko den mapaketa algoritmoaz gain embedding elebakarrak entrenatzeko erabiliko den algoritmoa ere zehaztu behar da. Guk kasu guztietan Word2Vec erabili dugu, zehazki Skip-gram bertsioa laginketa negatiboarekin, BiVec-entzat erabilitako parametro berdin-berdinekin (BiVec Skip-gramen hedapena da, eta parametro berdinak jasotzen ditu).

[1] https://github.com/facebookresearch/fastText/tree/master/alignment
[2] https://github.com/lmthang/bivec
[3] https://github.com/artetxem/vecmap

4.1.2 Hizkuntza pareak eta entrenamendu corpusak eta hiztegiak

Behin erabiliko diren metodoak zehaztuta, metodo horiek behar dituzten corpusak zehaztu beharko ditugu. Guk pareak zehaztean ordena kontuan hartuko dugu, hau da, C_1 eta C_2 bi corpus badira entrenamendurako $C_1 - C_2$ eta $C_2 - C_1$ pareen artean bereiziko dugu. Izan ere, embedding elebidunak lortzeko metodo batzuk simetrikoak diren arren (adibidez aldibereko metodoak edo Vecmap), beste batzuetan ordena garrantzitsua da, beraz embedding desberdinak lortuko dira corpus parearen ordena aldatzean.

Esan dugun bezala, hizkuntzen urruntasun linguistikoaren eta domeinu desberdintasunaren eragina aztertu nahi dugu. Halaber, aldibereko metodoak eta mapaketa-metodoak ere alderatu nahiko ditugu, beraz, kontu handia izan beharko dugu corpusak aukeratzean. Izan ere, mapaketa-metodoak erabiltzeko corpus elebakarrak lortzea nahikoa da, baina aldibereko metodoak erabili ahal izateko corpus elebidun lerrokatuak beharko ditugu.

ParaCrawl proiektuko corpus elebiduna erabiltzea erabaki dugu, zehazki finlandiar-ingeles pareko BiCleaner 3.0 bertsioa. Hemendik finlandiarreko corpus eta ingeleseko corpus bat lortu dugu, esaldi-mailan lerrokatuak egongo direnak. Corpus hauek FI_{PC} eta EN_{PC} deituko ditugu. Honez gain, wikipediako finlandiar, ingeles, aleman eta espainiar corpusak ere erabili ditugu, $\text{FI}_W, \text{EN}_W, \text{DE}_W$ eta ES_W deituko ditugunak. Azkenik, esperimentuetarako hurrengo corpus-pareak erabil ditugu:

- $\text{FI}_{PC} - \text{EN}_{PC}$
- $\text{FI}_W - \text{EN}_{PC}$
- $\text{DE}_W - \text{EN}_{PC}$
- $\text{ES}_W - \text{EN}_{PC}$
- $\text{EN}_W - \text{EN}_{PC}$

Hau da, helburu corpusa beti ParaCrawl corpus elebidunaren ingelesezko zatia izango da, eta jatorri corpusak wikipediako guztiak eta ParaCrawl-eko finlandiar zatia izango dira. Gainera, $\text{FI}_{PC} - \text{EN}_{PC}$ parearekin metodo sorta osoa erabili ahal izango dugu (mapaketa-metodoak + BiVec), baina gainontzeko pare guztiekin bakarrik mapaketa-metodoak erabiliko ditugu (corpusak ez direlako lerrokatuak izango). Ikus dezagun konfigurazio honen bidez kapituluaren hasieran azaldu ditugun hipotesi desberdinak egiaztatu ditzakegula:

Hizkuntza	Token kopurua
EN	2387 M
DE	860 M
ES	600 M
IT	523 M
FI	91 M

4.1 Taula: Erabilitako hizkuntzetako Wikipedia corpusen tamainak.

- $\text{FI}_{PC} - \text{EN}_{PC}$ parearekin lortutako emaitzen bidez mapaketa-metodo guztien propietateak alderatu ahal izango ditugu. Gainera, mapaketa-metodo desberdinak ere beste edozein parerekin lortutako emaitzen bidez alderatu ahal izango ditugu.

- $\text{FI}_{PC} - \text{EN}_{PC}$ eta $\text{FI}_{W} - \text{EN}_{PC}$ pareekin lortutako emaitzak alderatuz domeinuaren eragina aztertu ahal izango dugu.

- $*_{W} - \text{EN}_{PC}$ pareekin lortutako emaitzak alderatuz hizkuntzaren eragina aztertu ahal izango dugu.

Wikipedia corpusak lortzeko zehazki hurrengo prozedura jarraitu dugu: lehenik wikipediako raw dump-ak deskargatu ditugu web orritik [4], eta bigarrenik WikiExtractor [5] erabiliz testua erauzi dugu. Azkenik, Moses [6] erabiliz testua letra xehera pasatu eta tokenizatu dugu.

ParaCrawl finlandiar-ingeles corpuseko ingelesezko aldean 54.9 milioi token daude, eta wikipediako hizkuntza bakoitzeko corpusen tamainak 4.1 taulan agertzen dira.

Azkenik, mapaketa-metodo ez-gainbegiratuak egikaritzeko erabiliko diren entrenamendu hiztegiak ere zehaztu behar ditugu. Guk literaturan oso erabiliak diren bi hiztegi multzo erabili ditugu. Lehena, **Eparl** deitzen duguna, lehen aldiz Dinu et al.-ek sortua eta ondoren Artetxe et al. (2017) eta Artetxe et al. (2018a) lanetan hedatutakoa da, eta bere entrenamendu zatiak Europarl hitz lerrokaketetatik erauzitako 5000 hitzen itzulpenak ditu hizkuntza pare bakoitzeko. Bigarrena, guk **MUSE** deituko duguna, ere hizkuntza pare bakoitzeko 5000 hitzen itzulpenez osatua da, eta Conneau et al.-ek sortu zuten barne itzulpen erremintak erabiliz. Mapaketa-metodo gainbegiratu bakoitza bi aldiz egikarituko dugu, behin entrenamendu hiztegi bakoitzeko.

[4] https://dumps.wikimedia.org/
[5] https://github.com/attardi/wikiextractor
[6] https://github.com/moses-smt/mosesdecoder

4.1.3 Hubness metrikak

Atal honetan embedding elebidunetan hubnessa neurtzeko metrikak proposatuko ditugu. Demagun jatorri eta helburu hizkuntzetako embeddingen bektore multzoak X eta Y direla. Orduan, aurrekarien atalean azaldu dugun bezala, hubnessa neurtzeko bigarren hitzkuntzako $y \in Y$ hitz bakoitzeko $NN(y) = |\{x \in X : \forall y \in Y d(x,y) < d(x,y')\}|$ zenbakia kalkulatzen da, hau da, neurtzen da helburu hizkuntzako y hitza jatorri hizkuntzako zenbat x hitzen gertukoen auzokidea den (suposatzen dugu distantzietan berdintasunik ez dagoela, hau da, beti gertukoen auzokide bakar bat dagoela). Orduan $NN(y)$ balio altua duten hitzak hubak direla esaten dugu. Beraz, espazio osoaren hubness orokorra neurtzeko $NN(y)$ zenbaki hauetan oinarritutako metrikak erabiliko ditugu.

Literaturan kasu elebakarrean erabilia izan den metrika bat $NN(y)$ balioen frekuentzia banaketaren **skewness**a [7] da (Radovanović et al., 2010a). X aldagai aleatorio baten skewnessa honela definitzen da:

$$\gamma_1 = \mathbb{E}\left[\left(\frac{X-\mu}{\sigma}\right)^3\right],$$

eta banaketaren asimetria neurtzen du, hau da, balio positiboak izango ditu banaketak eskuinaldean pisu handia duenean. Adibidez, $NN(y)$ balio handia duten y asko existitzen badira, $NN(y)$-en frekuentzia banaketak skewness handia izango du.

Gure lehen esperimentuetan skewnessa erabili genuen, baina laster konturatu ginen hubness elebiduna neurtzeko arazo garrantzitsuak dituela metrika honek. Izan ere, skewnessa eskala-inbariantea da zatitzailean desbideratze tipikoa agertzeagatik, hau da, X eta kX aldagaien skewnessa berdin berdina da. Eskala inbariantza hau ez da egokia hubnessa neurtzeko; hau frogatzeko, bi adibide ikusiko ditugu:

- Suposa dezagun $|X| = |Y| = 100$ hitz ditugula hizkuntza bakoitzean, eta $NN(y)$ balioak horrela banatuta daudela: 97 y-k $NN(y) = 1$ balioa dute, batek $NN(y) = 3$, eta bik $NN(y) = 0$. Hau da, Y-ko 97 hitz X-ko hitz bakar baten gertukoen auzokideak dira, Y-ko hitz bat X-ko bi hitzen gertukoena da, eta ondorioz Y-ko bi hitz ez dira inoren gertukoenak (kontuan hartu behar da $\sum_{y \in Y} NN(y) = |X|$ izango dela beti, X jatorri hizkuntzako hitz bakoitzak Y-ko hitz bakar bat edukiko baitu gertukoentzat).

[7] *skewness* terminoa ingelesezko literaturan oso erabilia da, eta ez dugu itzultzeko modu argirik aurkitu, beraz itzuli gabe mailegutzat erabili dugu.

Orduan, frekuentzia banaketaren bataz bestekoa $\mu = \mathbb{E}(NN(y)) = 1$ (hau ere beti beteko da hubness balioen banaketetan) izango da eta skewnessa $\gamma_1 \approx 4.08$.

- Demagun orain X eta Y lehen bezelakoak direla, baina oraingoan $NN(y)$ balioen banaketa hurrengoa dela: 94 y-k $NN(y) = 1$ dute, 2 y-k $NN(y) = 3$, eta 4 y-k $NN(y) = 0$. Orduan kasu honetan bataz bestekoa ere $\mathbb{E}(NN(y)) = 1$ da, eta skewnessa $\gamma_1 = 2.88$.

Ikusi ditugun bi adibideen artean argi dago bigarrenak hubness altuagoa duela, baina skewnessa txikiagoa da bigarren kasuan, desbideratze tipikoa altuagoa izateagatik.

Beraz ikusi dugu skewnessa ez dela egokia hubnessa neurtzeko. Guk testuinguru elebidunera egokitutako hubness metrika bat proposatzen dugu. Metrika hau honela definitzen da: N ehuneko bat emanda, neurtzen dugu zein den helburu hizkuntzatik hartu behar den H_N hitzen ehuneko minimoa jatorri hizkuntzako hitzen $\%N$ estaltzeko. Esaten dugu jatorri hizkuntzako S hitz multzoa helburu hizkuntzako T hitz multzoak estaltzen duela S-ko hitz guztien itzulpenak (hau da, gertukoen auzokideak) T multzoan badaude.

Adibidez, $N = \%10$ bada, H_N zenbakiak adieraziko du gutxienez zenbat hitz (zein ehuneko) aukeratu behar ditugun helburu hizkuntzatik, jatorri hizkuntzako hitzen $\%10$-aren itzulpenak denak aukeratutako hitz hauen artean aurkitzeko.

Orduan, H_N balio altu batek esan nahiko du hubness txikiagoa dagoela, eta balio txikiek hubness altua adieraziko dute. Izan ere, helburu hizkuntzako hitz gutxi batzuk jatorri hizkuntzako hitz asko estaltzen badituzte, esan nahiko du helburuko hizkuntzako hitz gutxi batzuetan jatorriko hitz asko kontzentratu direla, hau da, hubness altua dagoela. Gainera, N parametro desberdinak probatuz hub banaketaren irudi argiagoa lortu dezakegu.

Ikusiko dugu orain $NN(y)$ balioak ezagututa H_N balioak nola kalkulatu daitezkeen. Aurkitu nahi dugu helburu hizkuntzako hitzen H_N ehuneko minimoa jatorri hizkuntzako $\%N$ hitz estaltzen dituena. Horretarako, Y helburu hizkuntzako hitzak $NN(y)$ balioaren arabera ordenatzen ditugu, $(y_1, ..., y_{|Y|})$. Jatorriko $\%N$ hitz estaltzeko, guztira $M = |X|N/100$ hitz estali behar dira. Gauzak honela, k balioa honela definitzen dugu: $k = \min\{j : \sum_{n=1}^{k} NN(y_n) \geq M\}$. Orduan, helburuko lehen k hitzek ($NN(y)$ balioen arabera ordenatuta) jatorriko M hitz estaliko dituzte, eta gainera k propietate hau betetzen duen balio txikiena izango da, helburuko hitzak $NN(y)$ balio handienetik txikienera ordenatu baiditugu. Ondorioz, bilatutako balioa k zenbakia ehunekora pasatuz lortuko dugu, hau da, $H_N = 100k/|Y|$.

Atal honetako esperimentuetan $N = \%100$ erabiliko dugu.

Deskribatutako H_N metrika $NN(y)$ balioen araberakoa denez, eta balio hauek NN edo CSLS atzipen bidez kalkulatu daitezkeenez, metrika hau ere aukeratutako atzipen metodoaren araberakoa da. Kapitulu honetako esperimentuetan NN atzipena erabili dugu hubnessa kalkulatzeko.

4.1.4 Ebaluazio metrikak

Embedding elebidunak ebaluatzeko erabiliko dugun metrika aurrekarien atalean azaldutako HEI izango da. Hiztegi indukzioa ebaluatu ahal izateko, urre patroi hiztegiak beharko ditugu. Horretarako, entrenatzeko erabiliko diren Eparl eta MUSE hiztegien ebaluazio zatiak erabiliko ditugu. Eparl-en ebaluazio zatiak Europarl hitz lerrokaketetatik erauzitako 1500 itzulpen ditu hizkuntza pare bakoitzeko, bost frekuentzia bandatan uniformeki banatuak. MUSE-ren ebaluazio zatia ere hizkuntza pare bakoitzeko 1500 itzulpenek osatzen dute, eta Conneau et al.-ek sortu zuten barne itzulpen erremintak erabiliz.

Mapaketa ez-gainbegiratu edo BiVec bidez lortutako embedding elebidunak bi ebaluazio hiztegiekin ebaluatuko dira, baina mapaketa gainbegiratu bidez lortutako embeddingak dagokien hiztegiarekin bakarrik ebaluatuko ditugu (adibidez MUSE-ren entrenamendu zatiarekin ikasi baldin bada mapaketa MUSE-ren ebaluazio zatiarekin ebaluatuko dugu embeddinga). Hau egiten dugu posible delako MUSE-ko entrenamendu zatiaren eta Eparl-eko ebaluazio zatiaren arteko ebakidura ez-hutsa izatea, edo alderantziz, eta ondorioz ebaluazioa modu gurutzatuan egiten bada gerta daiteke entrenamendu eta ebaluazio hiztegiek hitzak komunean edukitzea. Hau onartezina da, entrenamendu eta ebaluazio multzoen ebakidura ez-hutsa bada emaitza artifizialki onak lortuko baidira.

Gainera, aurrekarien kapituluan ikusi dugun bezala HEI ebaluazioa erabilitako atzipen metodoaren (NN edo CSLS) araberakoa da, beraz guk bi metodoekin egingo dugu ebaluazioa.

4.2 Emaitzak eta eztabaida

4.2 taulan esperimentuaren emaitzak aurkitu ditzakegu. Taulan emaitza asko jasotzen direnez, eztabaida hainbat zatitan banatuko dugu, eta atalez atal hasierako hipotesi bakoitza aztertuko dugu. Tauletan metodo gainbegiratuen kasuan metodo izenaren atzetik entrenatzeko erabilitako hiztegia agertzen da, Eparl edo MUSE.

Hizkuntza	Metodoa	Hub. NN (↑)	P@1 MUSE (↑)		P@1 Eparl (↑)	
		10%	NN	CSLS	NN	CSLS
$FI_{PC} - EN_{PC}$	Bivec	52.79	83.4	85.2	65.2	68.3
	Unsup	33.84	44.61	56.75	26.28	34.83
	Semisup	20.13	0.00	0.00	0.00	0.00
	Ident	33.83	44.54	56.68	26.34	34.76
	Sup MUSE	32.64	44.75	58.71	-	-
	Sup Eparl	32.15	-	-	25.86	37.17
	RCSLS MUSE	6.19	11.73	37.08	-	-
	RCSLS Eparl	3.44	-	-	4.41	18.14
$ES_W - EN_{PC}$	Unsup	22.74	0.07	0.07	0.00	0.00
	Semisup	21.90	0.00	0.00	0.00	0.00
	Ident	23.44	32.25	41.14	20.61	25.36
	Sup MUSE	23.71	36.46	46.44	-	-
	Sup Eparl	25.01	-	-	21.63	27.86
	RCSLS MUSE	8.30	34.15	39.44	-	-
	RCSLS Eparl	6.63	-	-	17.02	23.66
$DE_W - EN_{PC}$	Unsup	21.75	0.00	0.00	0.00	0.00
	Semisup	20.79	0.00	0.00	0.07	0.07
	Ident	22.92	23.54	30.95	16.25	21.30
	Sup MUSE	22.22	27.45	33.97	-	-
	Sup Eparl	24.06	-	-	17.69	23.10
	RCSLS MUSE	10.67	32.74	37.95	-	-
	RCSLS Eparl	6.75	-	-	13.94	20.51
$FI_W - EN_{PC}$	Unsup	19.98	0.00	0.00	0.00	0.00
	Semisup	21.05	0.00	0.00	0.00	0.00
	Ident	20.68	12.99	18.62	7.72	10.36
	Sup MUSE	18.67	20.90	30.74	-	-
	Sup Eparl	20.10	-	-	10.83	15.58
	RCSLS MUSE	4.21	19.22	26.79	-	-
	RCSLS Eparl	3.55	-	-	10.10	15.05
Hizkuntza	Metodoa	Hub. NN (↑)	P@1 EN-EN (↑)			
		%100	NN	CSLS		
$EN_W - EN_{PC}$	Unsup	21.65	0.07	0.07		
	Unsuport	22.56	0.07	0.07		
	Semisup	21.85	0.00	0.00		
	Ident	25.30	54.23	66.26		
	Sup	27.09	53.44	67.48		
	RCSLS	7.94	54.58	67.48		

4.2 Taula: Corpus pare eta metodo desberdinekin lortutako hubness eta P@1 balioak. ↑ geziak adierazten du balio altuagoak hobeak direla. Atzipen mota (NN edo CSLS) eta ebaluazio hiztegi (Eparl edo MUSE) bakoitzeko lortutako P@1 balioak aurkezten dira.

4.2.1 Domeinua eta hizkuntza

Metodoa	Parea	Hub. NN (↑)	P@1 Eparl (↑)		P@1 MUSE (↑)	
		10%	NN	CSLS	NN	CSLS
Ident	$FI_{PC} - EN_{PC}$	33.83	44.54	56.68	26.34	34.76
	$ES_{W} - EN_{PC}$	23.44	32.25	41.14	20.61	25.36
	$DE_{W} - EN_{PC}$	22.92	23.54	30.95	16.25	21.30
	$FI_{W} - EN_{PC}$	20.68	12.99	18.62	7.72	10.36
Sup MUSE	$FI_{PC} - EN_{PC}$	32.64	44.75	58.71	-	-
	$ES_{W} - EN_{PC}$	23.71	36.46	46.44	-	-
	$DE_{W} - EN_{PC}$	22.22	27.45	33.97	-	-
	$FI_{W} - EN_{PC}$	18.67	20.90	30.74	-	-
Sup Eparl	$FI_{PC} - EN_{PC}$	32.15	-	-	25.86	37.17
	$ES_{W} - EN_{PC}$	25.01	-	-	21.63	27.86
	$DE_{W} - EN_{PC}$	24.06	-	-	17.69	23.10
	$FI_{W} - EN_{PC}$	20.10	-	-	10.83	15.58

4.3 Taula: Metodo ez-gainbegiratu (iteratibo) eta gainbegiratu desberdinekin corpus pare bakoitzeko lortutako hubness eta P@1 balioak. ↑ geziak adierazten du balio altuagoak hobeak direla. Atzipen mota (NN edo CSLS) eta ebaluazio hiztegi (Eparl edo MUSE) bakoitzeko lortutako P@1 balioak aurkezten dira.

4.3 taulan ikusten dugu mapaketa-metodo bakoitzeko jatorri hizkuntza/domeinu desberdinekin lortutako emaitzak. Atera dezakegun ondorio argiena da, espero genuen bezala domeinuaren eragina oso handia dela. Izan ere, emaitza onenak, hubnessari begira, $FI_{PC} - EN_{PC}$ parearekin lortu dira, bi corpusak domeinu berekoak dituen kasu bakarra, naiz eta beste hizkuntza batzuk ingelesetik gertuago egon linguistikoki. Honez gain, FI-$_{W}$ - EN_{PC} eta FI_{PC}-EN_{PC} kasuen arteko desberdintasuna oso handia da, naiz eta hizkuntza parea berdina izan, batean domeinuak berdinak eta bestean desberdinak direlako.

Era berean, $*_{W} - EN_{PC}$ pareen emaitzetan ikusten dugu hizkuntzen desberdintasun linguistikoak ere eragina izan duela. Izan ere, hubness altuena $FI_{W} - EN_{PC}$ kasuan agertzen da, hain zuzen ere desberdintasun linguistiko handiena duen parean.

HEI ebaluazioari begira, emaitzak orokorrean hubnessarekin bat datoz: domeinuak ere eragin handia izan du, $FI_{PC} - EN_{PC}$ parean $FI_{W} - EN_{PC}$ parean baino emaitza askoz hobeak lortu baitira. Wikipediako corpusen artean ere finlandiarra izan da kasu okerrena, beraz, hizkuntza desberdintasunak ere eragina izan du HEIn.

Laburbilduz, esan dezakegu emaitza hauen arabera domeinu eta hizkuntza desberdintasu-

nek hubnessa eragiten dutela.

4.2.2 Mapaketa-metodoa, soluzio onaren ezaugarria

4.2 taulan corpus pare bakoitzeko mapaketa-metodo desberdinekin lortutako emaitzak agertzen dira.

Agian, taula honetako emaitza aipagarriena RCSLS metodoarena da. Izan ere, metodo honek hubness altuko embeddingak sortu ditu, baina, hala ere, kasu batuzetan hubness altuko embedding hauek HEI errendimendu ona izan dute, adibidez DE_W-EN_{PC} kasuan. Beraz ikusten dugu posible dela kasu batzuetan hubness altua izan arren HEI-n emaitza onak lortzea.

Hontaz gain, ezin dugu esan Vecmap gainbegiratuaren (ez-iteratiboa) eta ez-gainbegiratuaren (iteratiboa) arteko desberdintasun handia egon dela, ez hubnessaren eta ez HEI ebaluazioaren aldetik. Izan ere, kasu batzuetan (FI_{PC}-EN_{PC}) metodo gainbegiratuak hubness altuagoko soluzioa eman du, eta beste batzuetan (ES_W-EN_{PC}) kontrakoa gertatu da. Beraz, ikusten dugu Vecmapen pauso iteratiboak ez duela zertan hubness altua sortu behar.

Azkenik, hasieraketa txarreko metodo semi-gainbegiratuarekin lortutako emaitzek iradokitzen dute hubnessa ez dela soluzio on baten propietatea, kontrakoa baizik. Izan ere, metodo iteratiboari ausazko hasieraketa bat ematean optimo lokal txar batera iritsi da (hau HEI ebaluazioaren zutabean ikusten da, metodo hauek 0.0-ko P@1 lortu dute eta), eta gainera kasu honetan hubnessa orokorrean altuagoa izan da. Hau da, ikusten dugu bilaketa metodo iteratiboak ez dituela hubness altuko soluzioak bilatzen horiek onenak direlako, kontrakoa baizik, soluzio onak aurkitzean hubness txikiagoa agertu baita.

4.2.3 BiVec vs Mapaketak

4.4 taulan $FI_{PC} - EN_{PC}$ corpus parerako metodo guztiekin lortutako emaitzak jasotzen dira. Hemen alderatu ditzakegu BiVec aldibereko metodoarekin eta mapaketa-metodoekin lortutako emaitzak.

Argi ikusten dugu BiVec metodoaren bidez sortutako embeddingak onenak izan direla, bai hubness aldetik eta bai HEI aldetik, eta aldea oso handia izan dela gainera.

Emaitza horiek ikusita, pentsatu dezakegu hubness altua ez dela embedding elebidunen berezko propietate bat, embeddingak independenteki ikasi eta ondoren transformazio

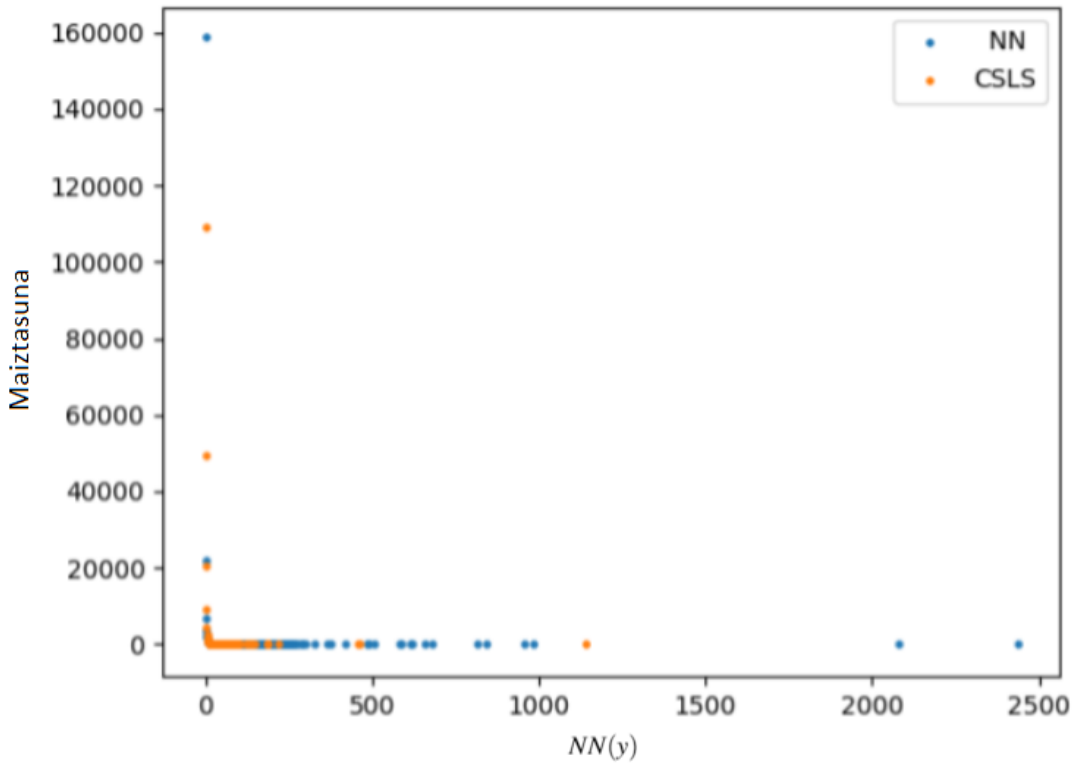
Hizkuntza	Metodoa	Hub. NN (↑)	P@1 MUSE (↑)		P@1 Eparl (↑)	
		10%	NN	CSLS	NN	CSLS
$FI_{PC} - EN_{PC}$	Bivec	52.79	83.4	85.2	65.2	68.3
	Unsup	33.84	44.61	56.75	26.28	34.83
	Semisup	20.13	0.00	0.00	0.00	0.00
	Ident	33.83	44.54	56.68	26.34	34.76
	Sup MUSE	32.64	44.75	58.71	-	-
	Sup Eparl	32.15	-	-	25.86	37.17
	RCSLS MUSE	6.19	11.73	37.08	-	-
	RCSLS Eparl	3.44	-	-	4.41	18.14

4.4 Taula: FI_{PC} - EN_{PC} corpus paralelo lerrokatuaren kasuan mapaketa-metodo desberdinekin lortutako hubness eta P@1 balioak. ↑ geziak adierazten du balio altuagoak hobeak direla. Atzipen mota (NN edo CSLS) eta ebaluazio hiztegi (Eparl edo MUSE) bakoitzeko lortutako P@1 balioak aurkezten dira.

linealen bitartez mapatzearen ondorioa baizik. Hala ere, kapitulu honetako esperimentuan bakarrik corpus pare bakar batekin erabili dugu BiVec metodoa, beraz honi buruz erauzitako edozein ondorio nahiko ahula izango da. Hurrengo kapituluan aldibereko eta mapaketa-metodoak alderatuko ditugu, ikusteko ea emaitza hauek berresten diren.

4.2.4 Hubnessa eta CSLS

4.2 taulan ikusten dugu kasu guztietan CSLS atzipena erabiltzean P@1 balio hobeak lortu direla. Are gehiago, NN eta CSLS atzipenekin lortutako balioen arteko desberdintasun handiena hubness oso handiko embedding batean agertu da ($FI_{PC} - EN_{PC}$ RCSLS kasuan). Hau zentzuzkoa da, CSLS neurria hubnessak HEI-n duen eragin negatiboari aurre egiteko diseinatu baizen. CSLS proposatu zuten autoreek hubnessa motibazio bezala erabili zuten arren, ez zuten aztertu neurri honek hubnessan duen eragina. Horregatik, interesgarria iruditu zaigu aztertzea ea espazioaren hubnessa txikitzen den NN atzipena ordez CSLS atzipena erabiltzean. Horretarako, FI_W-EN_{PC} corpus parearekin eta Ident mapaketa bidez lortutako embedding elebidunean helburu hizkuntzako hitz bakoitzeko $NN(y)$ balioak kalkulatu ditugu NN eta CSLS atzipena erabiltzen, eta balio horien frekuentzia totalen grafikak osatu ditugu. Emaitza 4.1 irudian agertzen da. Ikusten dugu CSLS atzipena erabiltzean espazioan orokorrean askoz hubness txikiagoa agertu dela: NN atzipenarekin existitzen dira y hitzak $NN(y) > 2000$ betetzen dutenak, hau da, 2000 hitzen gertukoen auzokideak direnak, eta CSLS atzipena erabiltzean $NN(y)$ balio altuena 1250 baino txikiagoa da.



4.1 Irudia: FI_W-EN_{PC} kasuan Ident mapaketa bidez lortutako embedding elebidunean $NN(y)$ balioen frekuentzia totalak, NN eta CSLS atzipena erabiltzean

Ikusitakoa kasu honetan gertatu den fenomeno isolatu bat ez dela frogatzeko, A eranskinean hainbat kasu desberdinetan lortzen diren grafikak aurkezten dira. Kasu guztietan CSLS atzipen bidez kalkulatutako hubnessa nabari baxuagoa da.

5. KAPITULUA

Mapaketa-metodoen mugak aldibereko metodoekin alderatuta

Aurreko kapituluan ikusi dugu BiVec aldibereko metodoak mapaketa-metodoek baino kalitate handiagoko embeddingak sortu dituela. Honez gain, ikusi dugu mapaketa-metodoek aldakortasun handia dutela; adibidez, $FI_{PC} - EN_{PC}$, $FI_W - EN_{PC}$ eta $ES_w - EN_{PC}$ pareekin oso emaitza desberdinak lortzen dira hubness eta HEI aldetik. Hala ere, ez dakigu zein den aldakortasun honen iturria.

Aurrekarien kapituluan ikusi dugu mapaketa-metodoen baliozkotasuna isomorfismo hipotesiaren mendekoa dela. Hipotesi honek dio independenteki hizkuntza bakoitzerako ikasitako embeddingek egitura antzekoak izango dituztela. Izan ere, egiturak oso desberdinak badira, ezinezkoa izango da modu ez-gainbegiratuan mapaketa bat aurkitzea, eta modu gainbegiratuan entrenatzeko seinale elebidun bat (adibidez hiztegi bat) badugu ere, zaila izango da transformazio lineal baten bidez oso desberdinak diren bi espazio lerrokatzea.

Halaber, azkenaldian autore batzuk isomorfismo hipotesi hau zalantzan jarri dute (Søgaard et al., 2018), frogatuz independenteki ikasitako embeddingak ez direla benetan isomorfoak, eta askotan isomorfo izatetik oso urrun daudela. Honez gain, embeddingen egituren arteko antzekotasuna neurtzeko autobalio antzekotasun metrika bat proposatu zuten. Autoreek ikusi zuten ere antzekotasun honen eta HEI ebaluazioaren artean korrelazio indartsua dagoela.

Guk pentsatu dugu isomorfismo hipotesia ez betetzea izan daitekeela mapaketa-metodoen aldakortasunaren iturria. Hau da, bi corpus elebakar erabiliz independenteki ikasitako

hitz-embeddingak egitura aldetik antzekoak direnean orduan erraza izango da mapaketa lineal baten bidez bi espazioak ondo lerrokatzea, eta mapaketa-metodoak arrakastatsuak izango dira. Independenteki ikasitako embeddingak isomorfo izatetik urrun daudenean, berriz, zaila izango da espazioak ondo lerrokatzea, eta ondorioz emaitza txarrak lortuko dira. Hau bat dator aurreko kapituluan ikusitakoarekin, zentzuzkoa baita pentsatzea domeinu edo hizkuntza desberdintasun handiak daudenean, independenteki ikasitako embeddingek egitura oso desberdinak edukiko dituztela.

Gainera, aurreko atalean ikusi dugu $FI_{PC} - EN_{PC}$ parearekin mapaketa-metodoak BiVec metodoak baino askoz emaitza hobea lortu duela, beraz ematen du aldibereko metodoak mapaketa-metodoek baino propietate hobeak dituela.

Orain deskribatutako hipotesiak egiaztatzeko eta aldibereko metodoen propietateak hobeto ulertzeko egindako esperimentuak eta ateratako ondorioak azalduko ditugu kapitulu honetan.

5.1 Esperimentuaren diseinua

Esperimentu honetan, 4. kapitulukoan bezala, hainbat corpus pare eta mapaketa desberdin erabiliz hitz-embedding elebidunak sortuko ditugu, eta emaitzak aztertuko ditugu hainbat metrika neurtuz. Oraingoan, hubnessa eta HEI ebaluazioa neurtzeaz gain, sortutako embeddingen autobalio antzekotasuna (isomorfismo neurria) ere kalkulatuko dugu.

Esperimentua egikaritu ahal izateko hurrengo faktoreak zehaztu behar ditugu:

1. Metodoak.
2. Hizkuntza pareak eta entrenamendu corpusak.
3. Hubness metrikak.
4. Ebaluazio metrikak.
5. Isometria metrikak.

5.1.1 Metodoak

Esperimentu sorta honetan hurrengo metodoak erabiliko ditugu:

Parea	Token kopurua
DE-EN	502 M
ES-EN	491 M
IT-EN	308 M
FI-EN	54 M

5.1 Taula: Erabilitako hizkuntzetako ParaCrawl corpusen tamainak.

- BiVec aldibereko metodoa, hurrengo parametroekin: 1e-5eko sub-sampling, 10eko laginketa negatiboa, 300eko bektore dimentsioa, eta 5 entrenamendu iterazio.
- VecMap ez-gainbegiratua, lehenetsitako parametroekin.

Metodo guztiak autoreek eskuragarri jarritako inplementazioekin egikaritu ditugu [1], [2], [3]. Emaitza taulak argiagoak eta ulergarriagoak izateko mapaketa-metodo bakarra erabili dugu oraingoan, Vecmap ez-gainbegiratua. Metodo hori ez-gainbegiratuen artean artearen egoeran dago, eta 4 kapituluan ikusi den bezala Vecmapen aldaera gainbegiratuaren eta ez-gainbegiratuaren artean ez dago desberdintasun handirik, lortutako embeddingen kalitatearen aldetik.

5.1.2 Hizkuntza pareak eta entrenamendu corpusak

Oraingoan BiVec eta mapaketa-metodoak alderatu nahi ditugunez, erabiliko ditugun corpus guztiak ParaCrawl [4] prokektutik ateratakoak izango dira, zehazki BiCleaner 3.0 bertsiotik. Erabiliko ditugun hizkuntza pareak finlandar-ingeles, espainiar-ingeles, aleman-ingeles eta italiar-ingeles izango dira. Corpus pare hauek $FI-EN$, $ES-EN$, $DE-EN$ eta $IT-EN$ bezala adieraziko ditugu kapitulu honetan. Helburu hizkuntza beti ingelesa izango da, eta kasu guztietan EN bezala adierazten dugun arren, garrantzitsua da jakitea kasu bakoitzean corpus hau desberdina izango dela. Hau da, ParaCrawl proiektuko corpusetan adibidez $FI-EN$ pareko eta $ES-EN$ pareko corpus paraleloen ingelesezko aldea ez da berdina izango, eta gainera parearen arabera tamaina aldetik desberdintasun nabariak izango dira. 5.1 taulan pare bakoitzeko ingeles aldeko token kopuruak adierazten dira.

[1] https://github.com/facebookresearch/fastText/tree/master/alignment
[2] https://github.com/lmthang/bivec
[3] https://github.com/artetxem/vecmap
[4] https://paracrawl.eu/releases.html

		Eig. sim. (↓)	Hub. NN (↑)		Hub. CSLS (↑)		P@1 Eparl (↑)		P@1 MUSE (↑)	
			10%	100%	10%	100%	NN	CSLS	NN	CSLS
FI-EN	Aldibereko ikasketa	**28.9**	**0.45**	**52.8**	**1.13**	**57.5**	**65.2**	**68.3**	**83.4**	**85.2**
	Mapaketa	115.9	0.12	33.8	0.38	46.1	26.3	34.8	44.6	56.8
ES-EN	Aldibereko ikasketa	**31.2**	**0.65**	**66.0**	**1.40**	**71.3**	**68.7**	**69.3**	**91.9**	**92.4**
	Mapaketa	47.8	0.58	63.1	1.31	69.1	65.4	67.0	87.1	89.0
DE-EN	Aldibereko ikasketa	**32.8**	0.58	**58.8**	1.29	**65.2**	**70.6**	**70.4**	**90.1**	**89.2**
	Mapaketa	39.4	**0.60**	58.7	**1.33**	64.8	65.3	66.4	82.4	83.1
IT-EN	Aldibereko ikasketa	**26.5**	**0.75**	**69.7**	**1.61**	**74.2**	**71.5**	**71.8**	**90.6**	**90.0**
	Mapaketa	43.9	0.65	63.9	1.53	70.8	64.1	67.2	84.4	85.9

5.2 Taula: Ebaluazio metrikak erabilitako embedding elebidun metodo bakoitzeko. Geziek adierazten dute ea balio baxuagoak (↓) edo altuagoak (↑) hobeak diren. P@1 ikurrak zehaztasuna adierazten du.

5.1.3 Hubness metrika

Hubnessa neurtzeko 4. kapituluan azaldutako H_N metrika erabiliko dugu. Oraingoan $N = \%10$ eta $N = \%100$ balioak erabiliko ditugu. Honez gain, metrika bakoitza NN eta CSLS atzipena erabiliz kalkulatuko dugu.

5.1.4 Ebaluazio metrikak

4. kapituluan bezala, embedding elebidunen kalitatea neurtzeko HEI ebaluazioa erabiliko da. Horretarako erabiliko ditugun urre-patroi ebaluazio-hiztegi sortak ere han deskribatutakoak izango dira, **Eparl** eta **MUSE**. Hitz-embedding elebidun bakoitza bi hiztegi hauekin (embeddingaren hizkuntza-pareari dagozkion bertsioekin) ebaluatuko da, eta bi emaitzak aurkeztuko dira. Gainera, bai NN eta bai CSLS atzipena erabiliz kalkulatuko ditugu zehaztasunak.

5.1.5 Isometria metrika

Isometria ebaluatzeko, aurrekarien kapituluan azaldutako autobalio antzekotasuna erabiliko dugu. Gogoratzen dugu autobalio antzekotasuna deitu arren, metrika honek benetan espazioen arteko desberdintasuna neurtzen duela, hau da, balio altuagoek espazioen egiturak desberdinagoak direla adierazten dutela.

5.2 Emaitzak

5.2 taulan esperimentuaren emaitzak jasotzen dira.

Autobalio antzekotasun metrikak erakusten du aldibereko ikasketak nabari isomorfikoagoak diren embeddingak lortzen dituela mapaketa-metodoarekin alderatuta, eta horrek iradokitzen du aldibereko metodoak erabiltzean egitura antzekoagoa duten embeddingak ikasten direla hizkuntza desberdinetarako. Aldi berean, aipagarria da lau hizkuntza pareen autobalio antzekotasuna oso antzekoa dela aldibereko ikasketa erabiltzean, 26.5 eta 32.8 artekoa. Mapaketa-metodoa erabiltzean, berriz, finlandiar-ingeles parearen isomorfismo neurria askoz baxuagoa da beste hizkuntza pareena baino, seguraski fi-en parearen desberdintasun tipologikoak handiagoak direlako beste pareenak baino, eta corpusen tamaina txikiagoa delako. Horrek iradokitzen du, mapaketa-metodoak ez bezala, aldibereko ikasketa gai dela hizkuntza dibergenteak modu egokian lerrokatzeko. Ondorioz, pentsa dezakegu aldibereko ikasketaren abantaila nagusia hizkuntzen desberdintasunen aurrean egonkorra izatea dela, mapaketa-metodoek baino aldakortasun askoz txikiagoa duelako.

Hubnessaren aldetik, gure emaitzek erakusten dute aldibereko ikasketak gutxiago sufritzen duela arazo hori, naiz eta desberdintasun handiak egon hizkuntza pareen artean. Adibidez, bi metodoek hubness antzekoa dute alemanaren kasuan, baina aldibereko ikasketak emaitza pixkat hobeak ematen ditu espainiar eta italiarraren kasuan, eta azkenik desberdintasuna oso nabaria da finlandiarraren kasuan. Esan bezala, honek iradokitzen du aldibereko metodoak askoz egonkorragoak direla dibergentzia linguistikoen aurrean. Aldi berean, ikusten dugu CSLS atzipenak hubness maila oso modu eraginkorrean murrizten duela, bereziki mapaketa-metodoen kasuan. Azken honek aurreko kapituluan ikusitakoa berresten du.

Azkenik, ikusten dugu aldibereko metodoek beti mapaketa-metodoek baino emaitza hobeak lortzen dituztela HEI egitean. Desberdintasun hori bereziki nabaria da finlandiar-ingeles kasuan (26.3% vs 65.2% NN eta Eparl hiztegiarekin), orain arte ikusitako patroia jarraitzen. Aldi berean, gure emaitzek erakusten dute CSLS mapaketa-metodoen kasuan dela eraginkorrena, eta kasu batzuetan eragin negatiboa izatera iristen dela aldibereko ikasketa erabiltzean. Fenomeno horren arrazoia izan daiteke aldibereko ikasketak hubness txikiagoa duela, eta CSLS hubnessa murrizteko diseinatutako neurria denez ulergarria da eraginkortasun txikiagoa izatea hubness fenomenoa ez denean agertzen (edo gutxiago agertzen denean).

5.2.1 Eztabaida

Gure analisiak erakusten du, corpus paraleloekin eta baldintza berdinetan entrenatzen direnean, aldibereko metodoek mapaketa-metodoek baino kalitate handiagoko embeddingak lortzen dituztela: aldibereko metodoekin lortutako embeddingak isomorfikoagoak dira, hubness txikiagoa dute, eta emaitza hobeak lortzen dituzte HEI atazan. Are gehiago, gure emaitzek frogatzen dute hizkuntzen arteko dibergentzia linguistikoak modu eraginkorrean mitigatzeko gai direla aldibereko metodoak, baina embeddingak independenteki ikasi eta ondoren mapatzen saiatzean ez dela berdina lortzen, eta ondorioz emaitza txarrak lortzen direla dibergentziak existitzen badira.

Garrantzitsua da aipatzea honek ez duela derrigorrean esan nahi aldibereko metodoak mapaketa-metodoak baino hobeak direla. Izan ere, gure ustez metodo familia bakoitzak helburu eta erabilera-kasu desberdinak dituzte, gainbegiraketa maila oso desberdinak eskatzen dituztelako (mapaketa-metodoek bakarrik corpus elebakarrak behar dituzte, corpus elebidun paraleloak baino askoz ugariagoak direnak). Hala ere, gure emaitzek frogatzen dute mapaketa-metodoek muga nabariak dituztela, eta, gure esperimentuek emandako ebidentziaren arabera, aldibereko metodoek ez dituztela muga hauek.

Arrazoi hauengatik argudiatzen dugu lan-lerro interesgarria izan litekeela muga hauek gainditzen dituzten metodo berriak garatzea. Partikularki, corpus elebakarrekin funtzionatzeko gai diren aldibereko metodoak garatzea bide interesgarria dela pentsatzen dugu.

6. KAPITULUA

Ondorioak eta etorkizunerako lana

Lan honek bi atal nagusi ditu. Lehen atalean (4. kapituluan) hitz-embedding elebidunetan agertzen den hubnessa aztertu dugu. Fenomeno horrek eragiten du dimentsio altuko espazioetan puntu gutxi batzuk beste askoren gertukoen auzokideak izatea, eta horrek eragin negatiboa du embedding elebidunen kalitatean. Gauzak horrela, hubnessa areagotu dezaketen hainbat faktore proposatu ditugu: domeinu desberdintasuna, dibergentzia linguistikoak, eta erabilitako metodoak. Faktore bakoitzaren eragina aztertzeko corpus pare eta metodo desberdinekin entrenatu ditugu embedding elebidunak, eta bakoitzaren hubnessa eta HEI errendimendua neurtu dugu. Ondorioztatu dugu entrenamenduan domeinu desberdineko corpusak erabiltzean hubness handia agertzen dela embedding elebidunetan, eta hizkuntzen arteko desberdintasun linguistikoek ere eragin nabaria dutela hubnessean. Erabilitako metodoei dagokionez, mapaketa-metodoen artean ez da desberdintasun handirik egon, baina ikusi dugu aldibereko ikasketa bidez lortutako embeddingek propietate askoz hobeak izan dituztela. Horrek eraman gaitu aldibereko metodoen eta mapaketa-metodoen arteko alderaketa sakonago bat egitera.

Mapaketa-metodoek ondo funtzionatzeko isomorfismo hipotesia betetzea garrantzitsua da. Hipotesi horrek dio independenteki ikasitako hitz-embeddingen egitura geometrikoak antzekoak izango direla, hau da, isomorfo izatetik gertu egongo direla. Azkenaldian autore batzuk (Søgaard et al., 2015) frogatu dute hipotesia ez dela guztiz betetzen, eta betetzen ez denean mapaketa-metodoek emaitza txarrak lortzen dituztela. Hala ere, ez dago argi ea ez-isomorfismo hori embeddingak independenteki ikastearen ondorioa den, edo dibergentzia linguistikoena, eta hortaz, embedding elebidunen arazo orokorragoa. Hori aztertzeko, lanaren bigarren atalean (5. kapituluan) aldibereko metodoak eta mapaketa-metodoak

alderatu dira. Kapitulu horretan, aurreko ataleko hubness eta HEI errendimendua neurtzeaz gain autobalio antzekotasuna ere neurtu dugu. Autobalio antzekotasuna isomorfismo metrika da, espazioen egituren arteko desberdintasuna neurtzen duena. Emaitzetan ikusi dugu, baldintza berdinetan eta corpus elebidun lerrokatuekin entrenatzen direnean, aldibereko ikasketa bidez sortutako embedding elebidunak mapaketa bidez sortutakoak baino isomorfoagoak direla, hubness txikiagoa dutela, eta HEI errendimendu hobea lortzen dutela. Ondorioztatu dugu ez-isomorfismoa embeddingak independenteki ikasi eta ondoren mapatzearen ondorio bat dela, eta aldibereko ikasketa gai izan dela dibergentzia linguistikoak gainditu eta ondo lerrokatutako embedding elebidunak sortzeko. Are gehiago, ikusi dugu aldibereko ikasketa mapaketa-metodoak baino askoz egonkorragoa izan dela hizkuntzen arteko dibergentzia linguistiko horien aurrean.

5. kapituluan aurkeztutako edukian oinarritutako artikulo bat ACL 2019 konferentzian argitaratuko da. [1]

6.1 Etorkizunerako lana

Ikusi dugu aldibereko ikasketak mapaketa-metodoek baino propietate askoz hobeak dituela. Dena den, horrek ez du esan nahi aldibereko ikasketa metodo hobea denik, bi metodo familiek gainbegiraketa maila guztiz desberdinak eskatzen baidituzte: aldibereko ikasketa erabili ahal izateko corpus elebidun lerrokatuak behar dira, eta mapaketa-metodoak erabiltzeko corpus elebakarrak nahikoa dira.

Sortutako embeddingen kalitatea corpus tamainaren mendekoa izaten da normalean, eta gaur egun hizkuntza askorentzat corpus elebakar handiak eskuragarri daude. Corpus elebidun lerrokatuak, berriz, askoz txikiagoak izaten dira, batez ere baliabide txikiko hizkuntza pareen kasuan. Honen ondorioz, kasu askotan ez da posible aldibereko ikasketa metodoak aplikatzea, eta ezin dira uztartu haien abantailak.

Gauzak horrela, pentsatzen dugu interesgarria izan daitekeela bi metodo familien abantailak konbinatzea, mapaketa-metodoen gainbegiraketa maila txikia mantenduz aldibereko ikasketaren abantailak uztartzeko gai diren metodo berriak garatuz. Etorkizunerako bide bereziki interesgarria iruditzen zaigu corpus elebakarrekin funtzionatzeko gai diren aldibereko metodoak garatzea.

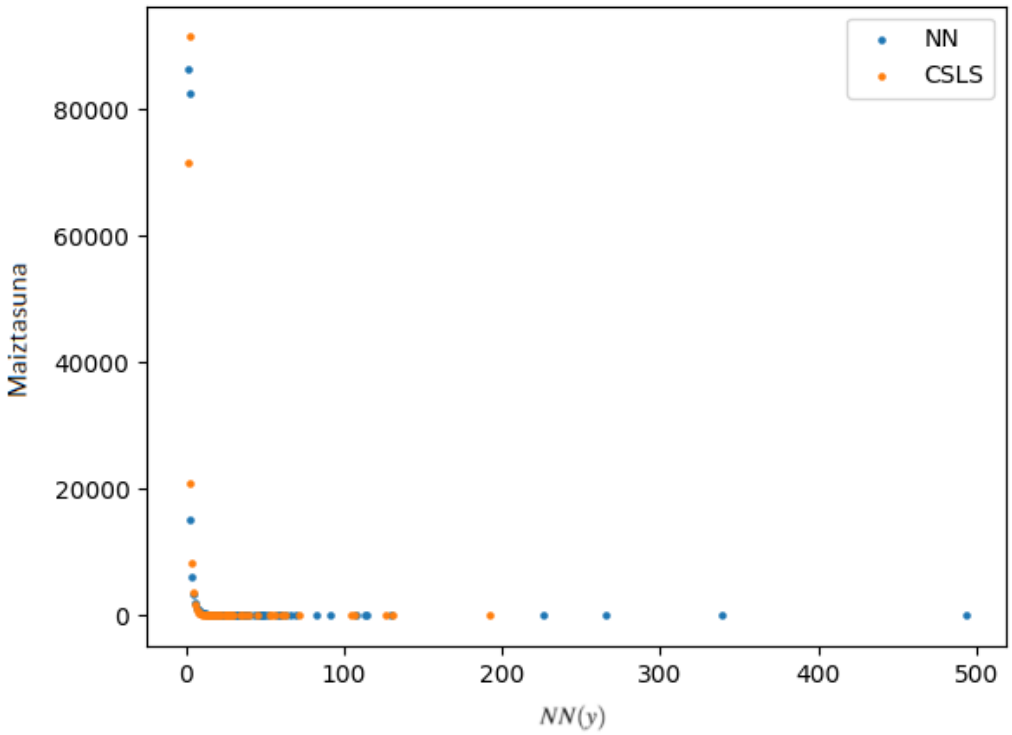
[1] ACL SCIE Class 1 konferentzia bat da. http://www.acl2019.org/EN/index.xhtml

Eranskinak

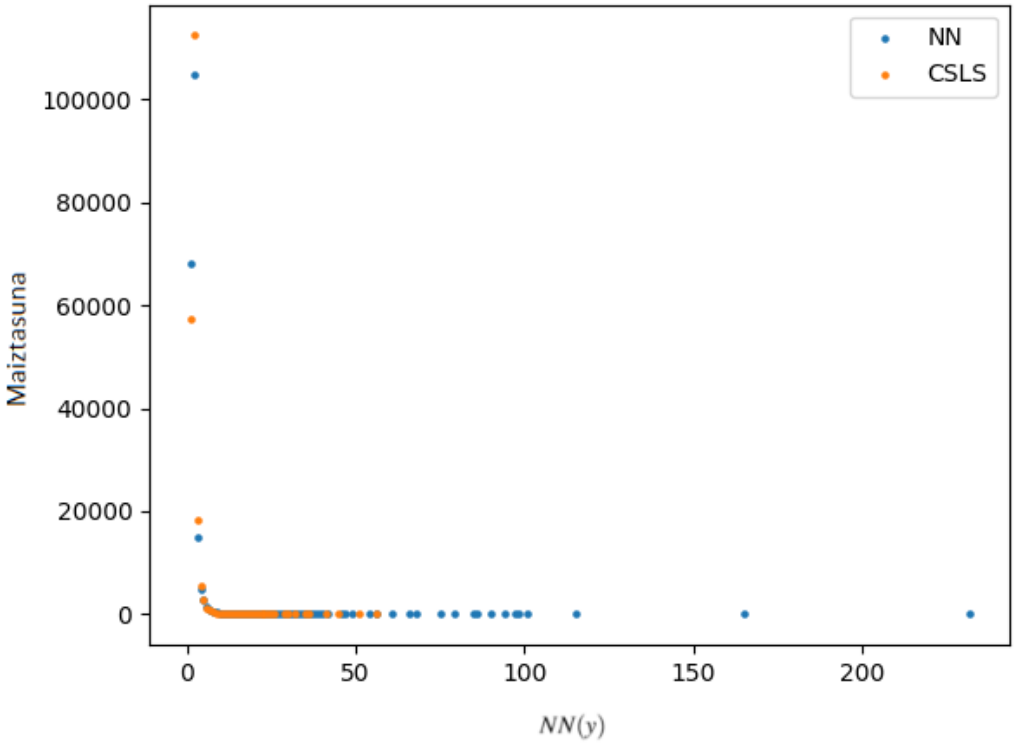
A. ERANSKINA

CSLS eta NN atzipenen alderaketa hubnessaren ikuspegitik

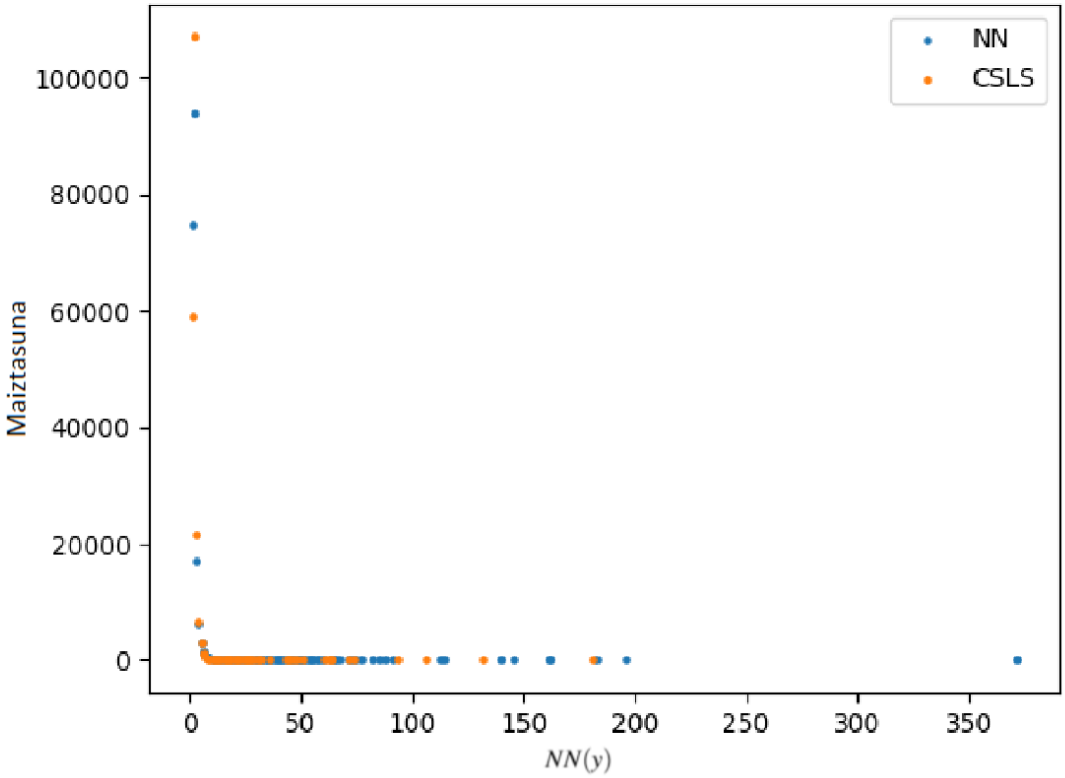
Eranskin honetan embedding elebidun desberdinetan NN eta CSLS atzipena erabiltzean agertzen diren hubnessak aurkezten dira. Horretarako, 4 kapituluaren bukaeran bezala atzipen mota bakoitzarekin kalkulatutako $NN(y)$ balioen frekuentzia totalen grafikak erabiltzen ditugu. A.1, A.2 eta A.3 irudietan agertzen dira emaitzak. Ikusten dugu kasu guztietan CSLS atzipen bidez kalkulatutako hubnessa nabari txikiagoa dela.



A.1 Irudia: DE_{PC}-EN_{PC} kasuan Vecmap gainbegiratu mapaketa bidez lortutako embedding elebidunean $NN(y)$ balioen frekuentzia totalak, NN eta CSLS atzipena erabiltzean.



A.2 Irudia: ES_{PC}-EN_{PC} kasuan bivec bidez lortutako embedding elebidunean $NN(y)$ balioen frekuentzia totalak, NN eta CSLS atzipena erabiltzean.



A.3 Irudia: IT_{PC}-EN_{PC} kasuan Vecmap gainbegiratu mapaketa bidez lortutako embedding elebidunean $NN(y)$ balioen frekuentzia totalak, NN eta CSLS atzipena erabiltzean.

Bibliografia

Artetxe, M., Labaka, G., and Agirre, E. (2017). Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462, Vancouver, Canada. Association for Computational Linguistics.

Artetxe, M., Labaka, G., and Agirre, E. (2018a). Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, pages 5012–5019.

Artetxe, M., Labaka, G., and Agirre, E. (2018b). A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798. Association for Computational Linguistics.

Conneau, A., Lample, G., Ranzato, M., Denoyer, L., and Jégou, H. (2018). Word translation without parallel data. In *Proceedings of the 6th International Conference on Learning Representations (ICLR 2018)*.

Dinu, G., Lazaridou, A., and Baroni, M. (2015). Improving zero-shot learning by mitigating the hubness problem. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), workshop track*.

Joulin, A., Bojanowski, P., Mikolov, T., Jégou, H., and Grave, E. (2018). Loss in translation: Learning bilingual word mapping with a retrieval criterion. In *Proc. of EMNLP 2018*, pages 2979–2984. Association for Computational LinguisticsACL.

Lazaridou, A., Dinu, G., and Baroni, M. (2015). Hubness and pollution: Delving into cross-space mapping for zero-shot learning. 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), pages 270–280. Association for Computational Linguistics.

Luong, T., Pham, H., and Manning, C. D. (2015). Bilingual word representations with monolingual quality in mind. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 151–159. Association for Computational Linguistics.

Miceli Barone, A. V. (2016). Towards cross-lingual distributed representations without parallel text trained with adversarial autoencoders. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 121–126. Association for Computational Linguistics.

Mikolov, T., Le, Q. V., and Sutskever, I. (2013a). Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119.

Radovanović, M., Nanopoulos, A., and Ivanović, M. (2010a). Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 11(Sep):2487–2531.

Radovanović, M., Nanopoulos, A., and Ivanović, M. (2010b). On the existence of obstinate results in vector space models. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 186–193. ACM.

Shigeto, Y., Suzuki, I., Hara, K., Shimbo, M., and Matsumoto, Y. (2015). Ridge regression, hubness, and zero-shot learning. In *Proceedings of the 2015th European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part I*, ECMLPKDD'15, pages 135–151, Switzerland. Springer.

Søgaard, A., Agić, v., Martínez Alonso, H., Plank, B., Bohnet, B., and Johannsen, A. (2015). Inverted indexing for cross-lingual NLP. 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)*, pages 1713–1722, Beijing, China. Association for Computational Linguistics.

Søgaard, A., Ruder, S., and Vulić, I. (2018). On the limitations of unsupervised bilingual dictionary induction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 778–788. Association for Computational Linguistics.

Zhang, M., Liu, Y., Luan, H., and Sun, M. (2017). Adversarial training for unsupervised bilingual lexicon induction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1959–1970, Vancouver, Canada. Association for Computational Linguistics.