

Informatika Ingeniaritzako Gradua

Konputazioa

Gradu Amaierako Lana

Juliet:

Eguneko albisteak lortzeko chatbot-a

Egilea

Iñigo Gabirondo López

2022

Informatika Ingeniaritzako Gradua

Konputazioa

Gradu Amaierako Lana

Juliet:

Eguneko albisteak lortzeko chatbot-a

Egilea

Iñigo Gabirondo López

Zuzendaria(k)

Gorka Azkune Galparsoro

Eneko Agirre Bengoa

Laburpena

Lan honen helburua eguneko albisteak emateko gai den Juliet chatbot-a sortzea eta Telegram aplikazioarekin integratzea da. Horretarako, hizkuntzaren prozesamenduko beste tresna batzuk ere erabili behar izan dira, adibidez, *scraping*, tokenizazioa, lematizazioa, kategoria gramatikalen etiketzea eta informazio-erazketa. Lan honi dagokionez, lehenik eta behin garapenean erabili diren teknika eta tresna desberdinen oinarri teorikoak azaldu dira. Ondoren, sistemaren arkitekturari eta honen funtzionamenduari buruzko argibideak eman dira. Azkenik, sortutako dialogo-sistemaren kalitatea neurtzeko, ebaluazio-protokolo bat definitu da, eta protokolo hori jarraituz, chatbot-aren ebaluazioa egin da. Ondorio nagusi bezala, dialogo-sistemak emaitza onak eman ditu eta, zenbait hobekuntza eginez gero, gure egunerokotasunerako oso erabilgarria izan daiteke. Gainera, garatutako arkitektura beste esparru batzuetan aplikagarria izateko zabal daiteke.

Gaien aurkibidea

Laburpena	i
Gaien aurkibidea	iii
Irudien aurkibidea	vii
Taulen aurkibidea	ix
1 Sarrera	1
1.1 Motibazioa	1
1.2 Chatbot-ak	2
1.3 Pertsonen eta konputagailuen arteko elkarrekintza (HCI)	3
1.4 Proiektuaren helburuak	4
1.5 Dokumentuaren egitura	5
2 Oinarri teorikoa	7
2.1 Dialogo-sistemak eta Rasa lan-ingurunea	7
2.1.1 Dialogo-sistemen oinarrizko kontzeptuak	7
2.1.2 Rasa lan-ingurunea	8
2.2 Informazio-erazketa eta Okapi BM25 algoritmoa	14
2.2.1 Okapi BM25 algoritmoa	16

iii

2.3	<i>Web scraping</i> teknika	18
2.4	Tokenizazioa, lematizazioa eta kategoria sintaktikoa	20
2.4.1	Tokenizazioa	20
2.4.2	Lematizazioa eta erro-bilaketa	21
2.4.3	Kategoria gramatikalen etiketatzea	22
3	Garatutako implementazioa	25
3.1	Sistemaren arkitektura	27
3.1.1	Rasa	28
3.1.2	Bilaketa-sistema	31
3.2	Funtzionalitateak eta hauen implementazioa	32
3.2.1	Berriak lortzea	32
3.2.2	Albiste bat bilatzea	33
3.2.3	Kategoria bateko albisteak lortzea	36
3.2.4	Albiste baten edukia lortzea	36
3.3	Implementazioa	38
4	Ebaluazioa eta lortutako emaitzak	39
4.1	Ebaluazio-protokoloa	39
4.2	Emaitzak	41
4.3	Eztabaida	45
5	Ondorioak eta etorkizuneko lana	49
5.1	Proiektuaren ondorioak	49
5.2	Ondorio pertsonalak	51
5.3	Etorkizunerako lana	52

Eranskinak

A	Proiektuaren helburuen dokumentua	57
A.1	Proiektuaren deskribapena eta helburuak	57
A.2	Proiektuaren plangintza	57
A.2.1	LDE diagrama	57
A.2.2	Lan - paketeak	57
A.2.3	Emangarriak	60
A.2.4	Mugarriak	60
A.2.5	<i>Gantt</i> diagrama	60
A.3	Lan metodologia	61
A.3.1	Bilerak	61
A.3.2	Planifikatutako ordutegiak	61
A.4	Bideragarritasuna	61
A.5	Arriskuak eta prebentzioak	62
A.5.1	Arriskuak	62
A.5.2	Prebentzioa	63
B	Ebaluazio galdetegia	65
C	Elkarrizketen adibideak	67
C.1	Intent sailkapen zuzena duen elkarrizketa	67
C.2	Intent sailkapen okerra duen elkarrizketa	68
C.3	Bilaketa zuzena duen elkarrizketa	69
C.4	Bilaketa okerra duen elkarrizketa	69
	Bibliografia	71

Irudien aurkibidea

1.1	ELIZA chatbot-aren elkarrizketa.	3
2.1	Esaldi baten prozesatzearen adibidea.	9
2.2	Laguntza intent-aren entrenamenduko datuak.	11
2.3	Erantzun bat sortzeko prozesua.	13
2.4	Informazio erauzketa eta BM25 prozesua.	15
2.5	Scraping prozesua.	19
2.6	Ingeleseko esaldi baten kategoria gramatikalen etiketatzea.	23
3.1	Albiste bat eskuratzeko elkarrizketa eredua.	26
3.2	Kategoria bateko albisteak eskuratzeko elkarrizketa eredua.	27
3.3	Juliet chatbot-aren arkitekturaren eskema orokorra.	28
3.4	Erabiltzailearen eta chatbot-aren arteko elkarrekintza.	30
3.5	Laguntza mezua erakusteko erregela.	31
3.6	Berriak lortzeko prozesua.	34
3.7	Albiste bat bilatzeko prozesua.	35
3.8	Kategoria bateko albisteak lortzeko prozesua.	37
3.9	Albiste baten edukia lortzeko prozesua.	38
4.1	“5 bilaketa desberdin egin dituzu, zenbatetan lortu dituzu albiste esanguratsusak?” galderaren emaitzak.	43

4.2	“Itzulitako albisteak alde batera utzita, izan ditudan elkarrizketa guztietan chatbot-ak zentzuzko gauzak esan ditu” galderaren emaitzak.	43
4.3	“Aurkitu nahi nituen albisteak Berriako web orrian baino azkarrago aurkitu ditut” galderaren emaitzak.	44
4.4	“Chatbot hau erabiliko nuke egunerokotasunean albisteak bilatzeko” galderaren emaitzak.	44
4.5	Intent sailkapen okerra duen elkarrizketa.	45
4.6	Bilaketa sistemaren emaitza okerra duen elkarrizketa.	47
4.7	Bi gaietara buruzko eskaera duen elkarrizketa.	48
A.1	LDE diagrama	58
A.2	<i>Gantt</i> diagrama	61
B.1	Ebaluaziorako erabili den galdetegia	66

Taulen aurkibidea

2.1	Ingelesaren tokenizazio irizpide desberdinen adibideak.	21
2.2	Lematizazioaren eta erro-bilaketaren konparaketa.	21
4.1	Ebaluazioaren emaitza kuantitatiboak.	42
A.1	Lan-pakete bakoitzari esleitutako ordu kopurua	58
A.2	Emangarrien entregatze datak	60

1. KAPITULUA

Sarrera

1.1 Motibazioa

Informazioa modu azkar, garbi eta antolatuan lortzea azken urteetako erronka handienetako batean bihurtu da [Singhal et al., 2001]. Datu kopurua hain da handia, non hori tratatzeko modu eraginkorrak diseinatzea, ondoren erabiltzaileari modu sinple batean emateko ezinbestekoa bilakatu den.

Bestetik, munduko edozein lekuetako gertakariak, zuzeneko edo zeharkako eragina dute gure bizitzetan, hori dela eta, munduan zehar gertatzen ari diren gauzei buruz ondo informatuta egotea garrantzia handikoa da.

Azken urteetan, *chatbot*-ek geroz eta ospe handiagoa irabazi dute, batik bat egindako azken aurrerapenek sortzen dituzten elkarrizketak geroz eta errealago eta koherenteagoak izatea lortu dutelako [yeung Shum et al., 2018]. Gaur egun, chatbot-ak telefono elkarrizketa bat izateko, hegaldi bat erreserbatzeko edota bezero baten galderei erantzuteko gai dira. Orduan, zergatik ez sortu chatbot bat egunean gertatu diren berriak lortzeko? Askotan, errazagoa da berrietaz jabetzea norbaitek kontatzen digunean.

Lan honetan, Berria egunkariko eguneko berriak emateko gai den Juliet izeneko chatbot-a garatu da eta erabiltzaileentzat modu eskuragarriago batean egon dadin, Telegram aplikazioarekin integratu da. Juliet chatbot-ak hiru elementu nagusi ditu: alde batetik, Berriko egunkaririk berriak eskuratu eta gordetzen dituen sistema, bestetik, lengoai naturala erabiliz albisteak eskuratzea ahalbidetzen duen modulua eta azkenik, erabiltzailearen eta

bilaketa-moduluaren arteko bitartekari lanak egiten dituen chatbot-a. Egindako lana ondo ulertzeko, lehenengo beharrezkoak diren oinarri teorikoak emango dira, ondoren sortutako arkitektura aztertze eta azkenik, lortutako emaitzak analizatzeko.

Arlo zientifikoari dagokionez, proiektu honek hizkuntzaren prozesamenduko arlo desberdinak ikutzen ditu, izan ere, berriak erabiltzaileari emateko, lehenengo berriak eskuratu, prozesatu eta gorde behar dira. Ondoren, berrien bilaketak modu optimoan egiteko sistema bat egin behar da, eta noski, erabiltzailearekin hitz egingo duen chatbot-a garatu behar da.

1.2 Chatbot-ak

Chatbot bat gizakion elkarrizketak (elkarrizketak ahotsez ala idatziz izan daitezke) prozesatu eta simulatzen dituen programa informatikoa da [Luo et al., 2022]. Modu horretan, erabiltzaileek beste pertsona erreal batekin hitz egiten egongo balira bezala erabil ditzakete aplikazio desberdinak. Gainera, chatbot-en konplexutasuna moldagarria da, izan ere, eskaera laburrei erantzuteko gai izatetik, erabiltzaileen datuak hartu eta laguntzaile birtual moduan atentzio pertsonalizatu bat emateraino programa baitaitezke.

Teknologia hau izen desberdinekin ere identifika daiteke, esate baterako, dialogo-sistema, elkarrizketa eragilea, elkarrizketarako interfazea, laguntzaile birtuala eta laguntzaile pertsonala.

Lehen dialogo-sistema, ELIZA chatbot-a izan zen [yeung Shum et al., 2018]. Erregeletan oinarrituta zegoen eta psikoanalista baten lana simulatzea zuen helburu. ELIZA chatbot-aren elkarrizketa bat ikus daiteke 1.1 Irudian. Elkarrizketa horretan, ELIZA-k galdera pertsonalak egiten dizkio erabiltzaileari eta hark bere arazoak kontatzen dizkio.

Hasieran, dialogo-sistemak ez ziren guztiz fidagarriak, baina azken hamarkadetan jasotako garapenen ondorioz, gaur egun ia webgune eta aplikazio guztietan aurki daiteke chatbot bat integratuta. Askotan ez gara ohartzen, baina web orri batean leiho bat azaltzen denean laguntza eskaintzen, edo gure mugikorrari ea bihar zer eguraldi izango dugun galdetzen diogunean, atzetik dagoena mota horretako sistema bat da.

Chatbot-en abantaila nagusia, erabiltzaileek bere helburua lortzeko lengoia naturala erabiltzeko aukera dutela da, gehienontzat erosoagoa eta sinpleagoa dena. Arestian esan bezala, gizakioi errazagoa egiten zaigu guk behar dugun informazioa beste norbaitek kontatzen badigu, gure kabuz bilatzea baino [Brandtzaeg and Følstad, 2017].


```

Welcome to

      EEEEE LL      IIII ZZZZZZ  AAAAA
      EE   LL      II   ZZ   AA  AA
      EEEEE LL      II   ZZZ  AAAAAA
      EE   LL      II   ZZ   AA  AA
      EEEEE LLLLL IIII ZZZZZZ  AA  AA

Eliza is a mock Rogerian psychotherapist.
The original program was described by Joseph Weizenbaum in 1966.
This implementation by Norbert Landsteiner 2005.

ELIZA: Is something troubling you ?
YOU:   Men are all alike.
ELIZA: What is the connection, do you suppose ?
YOU:   They're always bugging us about something or other.
ELIZA: Can you think of a specific example ?
YOU:   Well, my boyfriend made me come here.
ELIZA: Is it important to you that your boyfriend made you come here ?
YOU:   He says I'm depressed much of the time.
ELIZA: I am sorry to hear that you are depressed.
YOU:   It's true. I am unhappy.
ELIZA: Can you explain what made you unhappy ?
YOU:   █

```

1.1 Irudia: ELIZA chatbot-aren elkarrizketa.

Iturria: [Pérez, 2020]

Demagun enpresa batean, bezeroen galderari erantzuteko hamar langile dituztela. Langile kopurua mugatua denez, aldi berean erantzun daitezkeen galdera kopurua ere mugatua izango da, eta hau enpresarentzat ez da komenigarria. Bestalde, informazio guztia webgunean jartzea ere ez da beti aukera eraginkorra, bezeroek testu asko ikustean amore emateko arriskua dagoelako. Hori dela eta, chatbot bat erabiltzea aukera ona izan daiteke, alde batetik, erabiltzaileentzat bilaketak egitea erosoagoa izango delako, eta bestetik, eskaera asko aldi berean erantzuteko gai izango delako.

Beste alde batetik, chatbot-ek ere guk egindako eskaera eta galderetatik ikas dezakete, zerbitzu pertsonalizatu bat eman ahal izateko. Aurreko adibidera itzuliz, bezero batek eskaera bat egin behar duenean, chatbot-ak aldeztu aurretik bezero horrek egindako azken eskearatan oinarrituta dauden zenbait iradokizun eman ditzake.

1.3 Pertsonen eta konputagailuen arteko elkarrekintza (HCI)

Pertsonen eta konputagailuen arteko elkarrekintza, ordenagailuen interfazeen diseinua eta erabilera ikertzen duen informatikaren ikerketa arloa da [Sinha et al., 2010]. Gizakiok

aplikazio informatikoekin ditugun elkarrekintzak aztertzen dira bertan, elkarrekintza horiek izateko modu berriak sortu eta diseinatzeko.

Proiektu honetan, informatikaren arlo honek garrantzia handia du. Izan ere, sortzen ari garen notizien bilatzailea ez baldin bada erabiltzailearentzat erraza eta intuitiboa, berdin du atzetik dauden teknologia guztiak puntakoak diren ala ez, aplikazioa ez baita erabilgarria izango.

Hori guztia kontutan hartutan, lan hau garatzeko orduan, erabiltzaileek chatbot-arekin izango dituzten elkarrekintzei eta chatbot-aren implementazioari garrantzia berdina eman zaie, pertsona arrunt batek tresna hau erabiltzeko arazorik izan ez dezan.

1.4 Proiektuaren helburuak

Proiektu honen helburua eguneko berriak emateko gai den chatbot bat garatzea eta Telegram aplikazioarekin integratzea da.

Helburu nagusi horretara iristeko, hainbat azpi-ataza definitu dira, garapena modu iteratioboan egin ahal izateko:

- Chatbot-a garatzea:

Erabiltzaileen mezuen arabera, chatbot-ak erantzun bat ala bestea ematea, ala mota desberdinetako berriak bilatu behar ditu. Hori dela eta, dialogo-sistemak bi erronka nagusi aurkezten ditu: albisteak bilatzeko moduluarekin modu eraginkor batean komunikatzea (erabiltzaileen eskaerak azkarrago tratatzeko) eta erabiltzaileekin izan ditzakeen elkarrizketa posibleak ahalik eta sinpleenak izatea, chatbot-a erabiltzeko erraza izan dadin. Hori bermatzeko, lehenik eta behin chatbot-ei buruzko kontzeptuak ondo barneratu dira, ondoren Rasa *framework*-a erabiliz sortutako berrien bilatzailea erabiltzen duen chatbot bat garatzeko.

- Eguneko berriak lortzeko sistema diseinatzea eta implementatzea:

Chatbot-ak erabiltzaileari berriak emateko, lehenik eta behin hauek eskuratu, prozesatu eta gorde egin behar ditu. Ataza hau egunero egin behar denez, funtsezkoa da prozesu hori modu erraz eta azkar batean egingo duen tresna bat sortzea.

- Eguneko berriak bilatzeko sistema sortzea:

Sistema hau erabiltzeko erosoan izan dadin, erabiltzaileek bilaketak modu librean eta ahalik eta modu naturalenean egin behar dituzte. Ondorioz, hauen eskaerak hizkuntza librean izatea onartu behar da. Horretarako, gaur egun bilaketak egiteko existitzen diren algoritmoak aztertu dira, eta bilaketa-sistema eguneko berriak lortzeko sistemarekin integratu da.

- Chatbot-a ebaluatzea:

Elkarrizketak ebaluatzea ez da ataza erraza, dialogo berdina izateko modu zuzen asko baitaude. Hori dela eta, chatbot-aren errendimendua pertsonak ebaluatu behar dute, hau da, metrika automatikoki kalkulatu gabe. Ebaluazioa gauzatzeko, lehenik eta behin chatbot-a probatuko duten erabiltzaileek jarraitu behar duten protokolo bat definitu da, emaitzak homogeneoak eta adierazgarriak izan daitezen.

1.5 Dokumentuaren egitura

Dokumentu hau honela antolatzen da: Lan hau ulertzeko beharrezkoak diren oinarri teorikoak 2. kapituluaz azaltzen dira. Jarraian, 3. atalean, sortutako chatbot-aren arkitektura sakonki aztertzen da, egindako inplementazioaren eta honen funtzionamenduaren buruzko xehetasunak emanaz. Segidan, egindako ebaluazioa 4. atalean azaltzen da, proposatutako ebaluazio-protokoloa azalduz eta lortutako emaitzak analizatuz. Azkenik, 5. kapituluaz proiektu hau egin ostean izandako ondorioak eta etorkizunerako lanak aztertzen dira.

2. KAPITULUA

Oinarri teorikoa

Kapitulu honen helburua sortu den sistema ondo ulertzeko beharrezkoak diren oinarri teorikoak ematea da.

2.1 Dialogo-sistemak eta Rasa lan-ingurunea

Atal honen helburua, dialogo-sistemen oinarrizko kontzeptuak aztertzea eta Rasa lan-ingurunearen oinarrizko kontzeptuen azalpena ematea da.

2.1.1 Dialogo-sistemen oinarrizko kontzeptuak

Dialogo-sistema edo *chatbot*-a hizkuntza naturala erabiliz erabiltzaileekin elkarrizketa bat mantentzeko gai den softwarea da [Luo et al., 2022]. Izan ere, chatbot terminoa ingelesezko *chatterbot* hitzetik dator, eta hau pertsonekin hitz egiteko gai diren errobotak deskribatzeko proposatu zen.

Dialogo-sistemen arloan egindako aurrerapenak, horiek sortzen dituzten elkarrizketak ahalik eta errealenak izatean bideratu dira. Hori lortu ahal izateko, beharrezkoa da ataza desberdinetan hobekuntzak egiten jarraitzea.

Lehenik eta behin, *Natural Language Understanding* edo *NLU* ataza azpimarratu behar da. Horren bidez, chatbot bat erabiltzaile baten mezu bat prozesatzeko eta horren baitan koherentea den erantzun bat sortzeko gai da. Horretarako, chatbot-aren NLU modu-

lua erabiltzaileen mezuetatik *intent*-a eta entitateak lortzeko gai izan behar da. Intent-ek erabiltzaileak jarritako mezuaren asmoa edo intentzioa definitzen dute. Bestalde, entitateak (*entity*) erabiltzailearen mezuaren barnean egon daitezkeen hitz gakoak dira. Horrela, “Donostiko hondartzara joan nahi dut” bezalako mezu batean, “hondartzara joan” mezua-
ren intent-a izan daiteke, eta “Donosti” entitatea.

Bestetik, gizakiontzat oinarritzkoa bada ere, memoria eta testuingurua pisuzko faktoreak dira elkarrizketa batean. Izan ere, elkarrizketa baten testuinguruak, eta alde zuzenetik hitz egindako guztiak, informazio esanguratsua ematen dute zeharkako modu batean.

Azkenik, chatbot batek galderei ondo erantzuten erakusteko, elkarrizketa mota eta domeinu desberdinetako askotako kasu ugari prestatu behar dira eskuz. Gainera, kontzeptu berdina adierazteko modu asko existitzen direnez, ohikoa da erabiltzaileek dialogosistemak entrenamendu garaian ikusi ez duen esaldi bat erabiltzea. Ondorioz, garrantzitsua da chatbot-a erabiltzaileek emandako kasuetatik ikasteko gai izatea, kasu gehiagotan erantzun zuzen bat emateko gai izateko.

2.1.2 Rasa lan-ingurunea

Rasa errendimendu eta konplexutasun handiko adimen artifizialeko chatbot-ak sortzeko balio duen kode libreko lan-ingurune edo *framework* bat da [Bocklisch et al., 2017]. Eskalagarria, bereralbigarria eta kanal desberdinetan funtzionatuko duen chatbot-a modu erraz batean sor daiteke Rasa erabiliz.

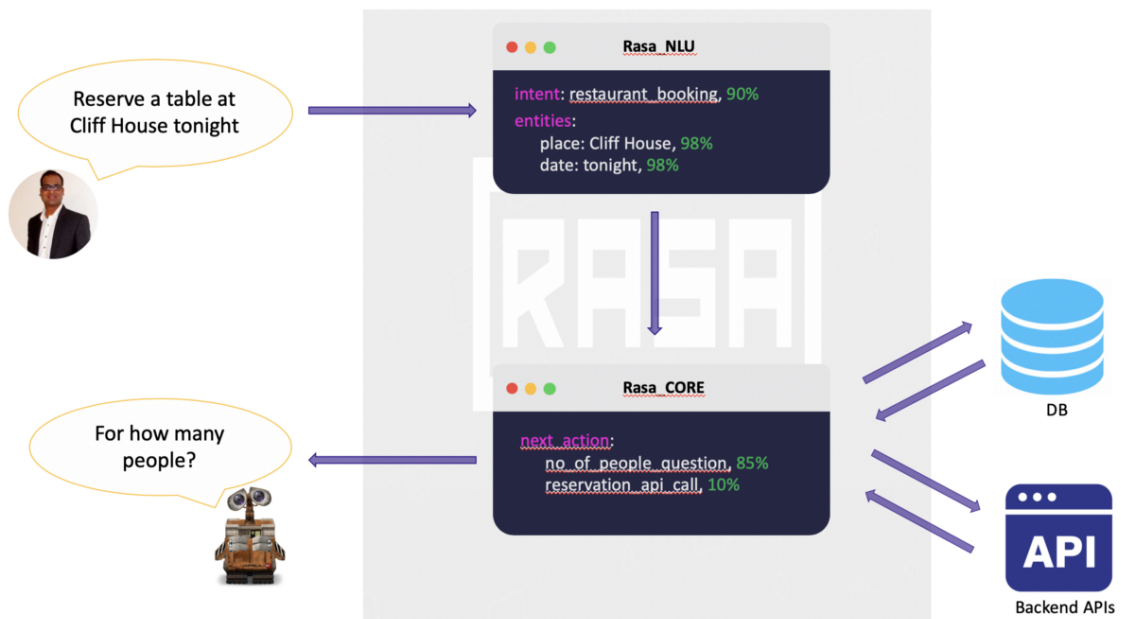
Ondorengoak dira Rasa-k eskaintzen dituen funtzionaltasun nagusiak:

- Mezuen esanahia lortzea:

Edozein hizkuntzetako testu hutsezko mezuetatik, modu lokalean entrenatzen den modelo baten bidez informazio egituratua (intent eta entitateak) lor daiteke. Horretaz gain, intent desberdinak eta alde zuzenetik entrenatutako eta pertsonalizatutako entitateak ere erabiltzeko aukera ematen du. Chatbot-aren NLU-a edozein domeinu eta erabilpen kasura molda daiteke.

- Elkarrizketa konplexuak maneiatzea:

Elkarrizketa bat aurrera doan heinean honen testuingurua gordetzen da, erantzun hobeak eman ahal izateko. Gainera, Rasa-k aparteko negozio logika inplementatzea eta dialogo-sistemarekin modu errazean integratzea ahalbidetzen du, elkarrizketak aberatsagoak izan daitezkeen.



2.1 Irudia: Esaldi baten prozesatzearen adibidea.

Iturria: [Palit, 2020]

- Ikasketa interaktiboa ahalbidetzea:

Entrenamendurako datuak sor daitezke erabiltzaileen elkarrizketetatik. Horretaz gain, chatbot-ak akatsen bat egiten duenean, hori konpontzeko atzeraelikadura eman daiteke. Gainera, garatutako dialogo-sistema, erabiltzaileen artean modu errazean partekatu daiteke, esteka baten bidez. Modu horretan, chatbot-aren ebaluazioa egitea asko sinplifikatzen da.

- Chatbot-a normalean erabiltzen diren kanalekin konektatzea:

Laguntzaile birtuala *Google Home*, *Telegram*, *Facebook* edo *Slack* bezalako kanaletan funtzionatzeko konfiguratu daiteke. Chatbot bakarrak kanal desberdinetan eman dezake zerbitzua.

- Software ingeneritzako praktika onak mantentzea:

Rasa-ren bidez chatbot-a sortzean, honen implementazioa modulo desberdinetan banatzen da, programazio modularra eta iteratiboa ahalbidetuz. Hortaz gain ere, dialogo-sistemaren bertsio desberdinak gordetzeko, eta hauen artean aurrera eta atzera egiteko aukera ematen du.

Rasa-k bi atal nagusi ditu: Rasa NLU eta Rasa core. Rasa NLU intent sailkapena eta entitateak ateratzeko balio duen hizkuntzaren prozesamendurako tresna bat da. Mezu bat

iristen denean, liburutegi honek mezua prozesatzen du, bertatik informazio egituratua sortzen du eta hurrengo atalari bidaltzen dio. Beste modu batera esanda, erabiltzaileen mezuek “ulertzen” dituen atala da. Rasa Core atalak, Rasa NLU-tik jasotako informazio egituratua hartu, eta horiei erantzuteko chatbot-ak exekutatu behar duen hurrengo akzioa hautatzen du. Akzio bat, testu hutsezko mezu bat bidaltzea, datu-base bati dei bat egitea eta emaitzak bidaltzea edo API bati deiak egitea izan daiteke.

Rasa-ren atalak eta horien arteko elkarrekintza 2.1 Irudian ikus daitezke. Bertan, erabiltzaileak jabetxe batean mahai bat erresebartzeko eskaera egiten dio chatbot-ari. Erabiltzailearen mezua NLU modulura iristean, modulu horrek eskaera prozesatzen du mezuaren intent-a eta entitateak lortzeko. Horren ostean, NLU moduluak informazio hori Core atalari bidaltzen dio, eta horrek, jasotako datuen arabera chatbot-ak exekutatu behar duen hurrengo ekintza zein izango den erabakitzen du. Segidan, ekintza hori exekutatzen da (pauso horretan Rasa-tik kanpo dauden API eta datubaseei deiak egiten zaizkie), ondoren erantzun bat sortzen da eta azkenik, erantzuna erabiltzaileari bidaltzen zaio.

Laburbilduz, Rasa NLU-ren eginkizuna erabiltzailearen mezuek interpretatzea eta bertatik informazioa esanguratsua lortzea da, eta Rasa Core-ena, berriz, chatbot-ak emango duen hurrengo pausua zein den erabakitzea da. Jarraian bi elementu horien zehaztasunak emango dira eta ondoren mezu bat prozesatzeko bi atal horien arteko elkarrekintza azalduko da.

Rasa NLU

Lehen aipatu den bezala, Rasa NLU erabiltzaileen mezuetatik informazio egituratua lortzen den liburutegia da. Horrela, chatbot-ari komunikazio kanaletik mezu bat iristen zaion bakoitzeko, honi intent bat esleitu eta bertan egon daitezkeen entitateak lortuko ditu.

Bestalde, entitateen zein Rasa-tik kanpoko informazio gordetzeko, *slot* izeneko aldagaiak erabiltzen dira. Giltza-balio moduko aldagaiak dira, eta funtsean, goi-mailako programazio-lengoaiek eskaintzen dituzten hiztegien eginkizun bera betetzen dute. Ohikoa da entitate bakoitzeko *slot* bat definitzea, entitate horien balioak *slot*-aren bidez atzitu ahal izateko.

Intent sailkapena eta entitateen detekzioa egiten ikasiko duen modelo bat lortzeko, *NLU pipeline* bat definitu behar da. *Pipeline* batek, osagai desberdinak ditu, eta modu sekuentzian lan egiten dute, erabiltzaileen mezuek informazio egituratuan bihurtzeko. Modu horretan, entrenamenduko eta erabiltzaileek sarrera moduan emandako esaldiek, pipeline-ean zehaztutako kate osoa egiten dute, bertatik informazioa lortuz. Osagaien artean, tes-


```
- intent: ask_for_help
  examples: |
    - Laguntza
    - Laguntza behar dut
    - Laguntza eskatu nahi dut
    - Laguntza ondo etorriko litzaidake
```

2.2 Irudia: Laguntza intent-aren entrenamenduko datuak.

tu preprozesatzaileak, intent sailkatzaileak, izendun entitateen identifikatzaileak (NER)... bezalako elementuak aurki daitezke. Gainera, hizkuntzaren prozesamenduko munduan erabiltzen diren beste tresna batzuk zuzenean integratzeko aukera ere ematen du, adibidez SpaCy-ko *word embedding*-ak edo tokenizatzaileak [Honnibal and Montani, 2017].

Horrela, gure chatbot-ak intent-at eta entitateak sailkatzen ikastea nahi badugu, entrenamenduko datuak eman behar dira. Erabiliko den intent bakoitzeko, intent hori adierazten duten hainbat esaldi zehaztu behar dira. Laguntza intent-aren hainbat entrenamenduko adibide ikus daitezke 2.2 Irudian. Adibide horiek YAML formatuan idatzita daude eta guztiek laguntza eskatzea dute helburu.

Kasu horretan, sarrera mezua adibidean definitutako esaldien antzekoa bada, mezuaren intent-a laguntza bezala sailkatuko da. Entitateak lortzeko ere, entrenamenduko adibideak eman behar dira.

Oinarrizko adibideen gain, entitateak lortzearen eta intent sailkapenaren atazak errazteko asmoz, hainbat datu-egitura erabil daitezke entrenamenduko datu bezala:

- Espresio erregularrak:

Rasa-k espresio erregularrak definitzeko aukera ematen du, eta horien bidez, NLU modulua bi ataza nagusiak egin daitezke.

- Bilaketa-taulak (*lookup-tables*):

Bilaketa-taulak patroiak sortzeko erabiltzen diren hitzen zerrendak dira. Demagun Europako herrialdeak entitate bezala identifikatu nahi ditugula. Herrialde kopurua finitua denez, hauen guztien izenak zerrenda batean jarri ditzakegu, eta horrela, sarrera mezuan zerrendan dagoen herrialde bat azaltzen bada, NLU moduluak hori herrialde bat dela identifikatuko du eta entitate moduan gordeko du. Bilaketa-taulak intent-ak sailkatzeko ere balio dute.

- Sinonimoak:

Sinonimoek mezuetatik ateratako entitateak balio batera lotzen dituzte. Adibidez, demagun erabiltzailearen mezuan “Portugalen” entitatea lortu dela. Hau deklinatuta dagoenez, ondoren erabiltzeko arazoak eman ditzake. Sinonimo bat definituz, NLU moduluak lehenengo “Portugalen” entitatea lortuko luke, ondoren entitate hori “Portugal” baliora esleituz.

Rasa core

Rasa core moduluaren helburua, Rasa NLU-tik intent eta entitateak lortzea, eta horietan oinarrituz chatbot-ak egin behar duen akzioa iragartzea da. Akzio hauek eginez, erabiltzaileei irteera mezuak bidaliko zaizkie.

Rasa-ren bidez egindako chatbot batek, bi motatako akzioak egin ditzake:

- Erantzunak (*responses*).

Akzio sinpleena da, izan ere, erantzunak dialogo-sistemak erabiltzaileari zuzenean bidaltzen dizkion mezuak dira. Mezu horiek orokorrean testu hutsa baino ez dute izaten, baina irudiak, botoiak... bezalako elementuak ere izan ditzakete.

- Akzio pertsonalizatuak (*custom actions*).

Akzio pertsonalizatuek, intent konkretu bateko mezua iristen denean, garatzaileak nahi duen kode zati bat exekutatzeari ahalbidetzen dute. Horien bidez, API bati deiak egitea edo datubase batean kontsultatzea posible da, hau da, negozio logika zabalago eta aberatsago bat sortzea ahalbidetzen dute. Akzio horiek idazteko programazio lengoia Python3 da [Van Rossum and Drake, 2009]. Akzio pertsonalizatuak chatbot-arekin etengabe komunikatzen ari den aparteko zerbitzari batean exekutatzen dira.

Dialogo-sistemak zein akzio exekutatu behar duen eta ikasketa garaian ikusi ez diren elkarrizketa-ereduak orokortzen ikas dezake, *stories* edo istorioak definituz. Istorioek elkarrizketa-patroiak errepresentatzen dituzte, non erabiltzaileen intent-en eta chatbot-aren akzioen arteko interakzioak zehazten diren. Horien bidez, chatbot-ak izan ditzakeen elkarrizketen eredu desberdinak idatz daitezke, hala nola, hegaldi bat erreserbatzeko istorioa, hondartzara joateko istorioa edo errore-kasuak kudeatzeko istorioa. Istorioekin oso

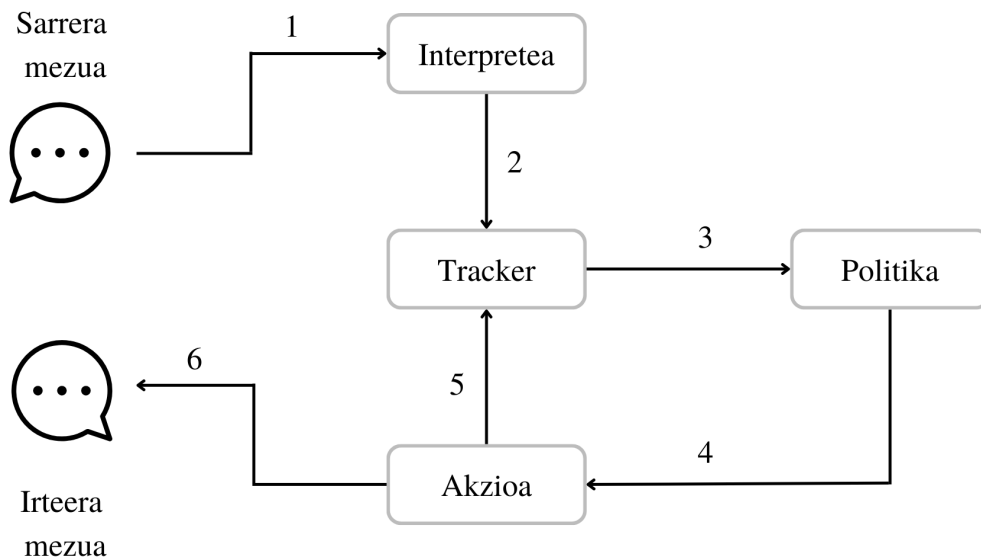
lotuta, erregelak ere erabil daitezke interakzio laburragoak definitzeko. Elkarrizketaren zati konkretu batzuek beti bide berdina jarraitzen dutela ziurtatzea da erregelen helburua.

Bestalde, istorioak eraginkorrak izan daitezten, beharrezkoa da elkarrizketaren nondik norakoak gordetzen joatea. Hori egin ahal izateko, *tracker* elementua dago. Elkarrizketa aurrera joan ahala, mezu bakoitzaren intent eta entitateak, eta mezu horren ondorioz exekututako akzioak gordetzen joango dira. Horrela, tracker-aren bidez, elkarrizketaren uneko egoera zein den eta azken mezuak zeintzuk izan diren jakin ahal izango da.

Politiken (*policies*) bidez, erabiltzaileak bidalitako mezu bakoitzaren ostean dialogo-sistemak zein akzio exekutatu behar duen iragartzen da. Pipeline-ekin gertatzen den antzera, politika desberdinak sor daitezke, erabiltzeko prest dauden tresna desberdinak erabiliz eta hauek beraien artean konbinatuz. Kasu honetan, *machine learning* eta erregelen bidezko politikak erabiltzen dira aldi berean.

Mezuak prozesatzea

Hau guztia definituta, garrantzitsua da Rasa-k sarrera mezu bat prozesatzeko eta horren baitan erantzun bat sortzeko jarraitzen duen prozesua ulertzea:



2.3 Irudia: Erantzun bat sortzeko prozesua.

1. Erabiltzaileak mezu bat idazten dio chatbot-ari komunikazio kanalaren bitartez eta hori interpretari iristen zaio.

2. Interpretarean atalean, Rasa NLU liburutegiak mezua prozesatu eta bertatik baliagarria den informazioa lortzen du, adibidez, intent-a, entitate posibleak, sinonimoak... Informazio guzti hori tracker-ari bidaltzen dio. Prozesu hontako beste elementu guztiak Rasa Core atalean gertatzen dira.
3. Tracker-aren eginkizuna elkarrizketaren istorio osoa memorian gordetzea da. Hori eginez, chatbot-ak uneoro daki elkarrizketaren zein puntutan dagoen, ondoren erabakiak hartu ahal izateko.
4. Horren ostean, politikak elkarrizketaren istorian oinarrituz, exekutatu den hurrengo akzioa zein den erabakitzen du.
5. Akzioa exekutatu aurretik, tracker-ari elkarrizketaren egoera eskatzen zaio, bertan gordetzen den informazioa (azken mezuaren intent-a, lortutako entitateak...) baliagarria delako.
6. Azkenik, politikak zehaztutako akzioa exekutatzen da, eta erabiltzaileari irteera mezua bidaltzen zaio. Akzioa erantzun bat edo akzio pertsonalizatu bat izan daiteke.

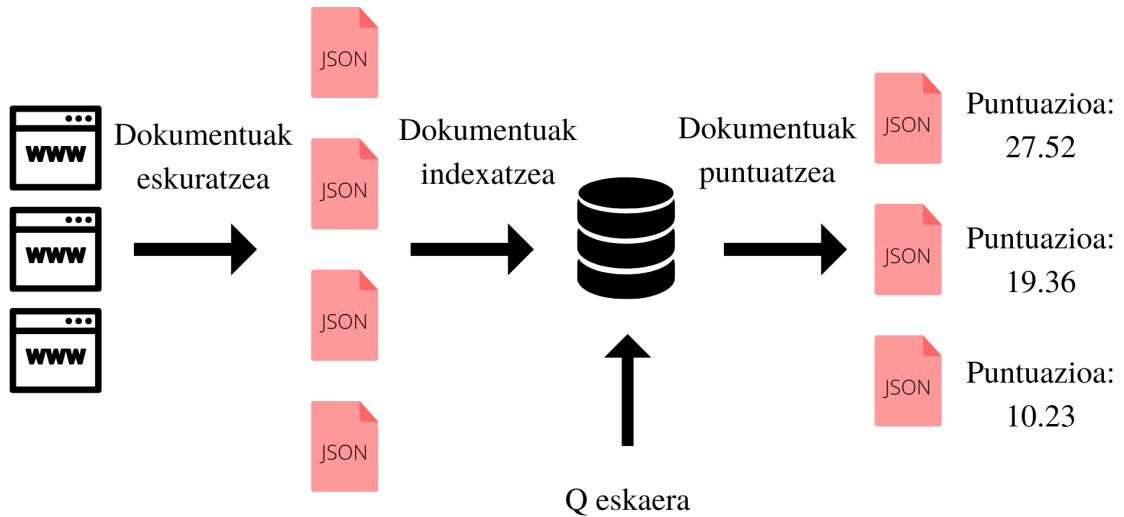
[2.3](#) Irudiak erantzun bat sortzeko prozesua erakusten du.

2.2 Informazio-erazketa eta Okapi BM25 algoritmoa

Informazio-erazketa (ingelesezko *information retrieval* edo *IR*), dokumentuz osaturiko datubase batetik, informazio eskaera batetik adierazgarrienak diren dokumentuak lortzean datza. Orokorrean, datubasean egoten diren dokumentuek testu hutsezko informazio ez egituratua gordetzen dute. Horrela, eskaera bat iristen denean, datubaseko dokumentu bakoitzari horren garrantzia neurtzen duen funtzio baten bidez puntuazio bat esleitzen zaio, ondoren puntuazio altuena lortu duten dokumentuak itzuli ahal izateko.

Funtsean, software horien helburua ez da eskaera baten erantzuna zuzenean ematea, baizik eta eskaera horren erantzuna izateko probabilitate altuena duten dokumentuak itzultzea. *IR* sistemak egunero erabiltzen ditugu konturatu gabe, izan ere, sistema horien adibide garbienak edozein nabigatzailetan erabiltzen ditugun bilatzaileak dira, non bilatu nahi dugun informazioaren hitz gakoak idatziz, bilatzaileak guk emandako sarrerakiko garrantzitsuenak diren webguneak itzultzen dituen.

Informazio erazketako sistema bat sortzeko ondorengo elementuak behar dira:



2.4 Irudia: Informazio erazketa eta BM25 prozesua.

- Dokumentuen eskuratzea:

Lehenik eta behin, oso garrantzitsua da interesatzen zaigun domeinuari buruzko dokumentu asko lortzea eta datubase batean gordetzea. Erabiltzaileen eskaerak oso zabalak izan daitezke, eta lortutako dokumentu kopurua ez bada behar bezain handia, litekeena da gure bilatzaileak eskaeretikiko garrantzitsuak diren dokumentuak ez aurkitzea. Dokumentuen datubaseari indizea deitzen zaio, eta dokumentu bakoitza datubasean gordetzeko prozesuari, berriz, indexatzea.

- Dokumentuen errepresentazioa:

Orokorrean, lortuko ditugun dokumentuen informazioa ez da egituratuta egongo. Horregatik, interesgarria izan daiteke dokumentu bakoitza ezaugarri desberdinetatik errepresentatzea. Adibidez, liburuak indexatuz gero, zentzuzkoa izango litzaiteke liburuaren izenburua, idazlea, laburpena, metadatuak eta liburu bera gordetzea.

- Eskaerak prozesatzea:

Eskaerak, informazio beharra adierazten duten testu kateak dira, adibidez, nabigatzaile baten bilatzailean erabiltzen dira testu kateak. Modu horretan, IR sistemara eskaera bat iristen denean, dokumentu bakoitzari puntuazio bat esleitzen zaio, eskaeretikiko garrantzia neurtzen duen funtzio baten bidez. Hori egin ostean, puntuazioak

zio altuenak lortu dituzten dokumentuak bidaliko zaizkio erabiltzaileari. Software hauen indizeak oso handiak izaten dira, baina hala ere eskaeren erantzunak oso denbora laburrean eman behar dituzte. Horretarako, beharrezkoa da dokumentuak puntuatzeko erabiltzen den funtzioa eraginkorra izatea.

Azkenik, informazio erauzketaren prozesu orokorra 2.4 Irudian ikus daiteke. Lehenengo, webguneetatik informazioa lortu, informazio hori JSON fitxategietan idatzi eta fitxategi horiek indizean gorde dira. Ondoren, erabiltzaile batek eskaera bat egiten duenean, indizeko dokumentuei puntuazio bat esleitu zaie eta azkenik, puntuazio altuena lortu duten hiru dokumentuak itzuli zaizkio erabiltzaileari.

2.2.1 Okapi BM25 algoritmoa

Okapi BM25 informazio erauzketan erabiltzen den sailkapen funtzio bat da, eta dokumentuei garrantzia esleitzeko balio du [Robertson and Zaragoza, 2009]. Beste modu batera esanda, indize bateko dokumentuei Q eskaera batetik duten puntuazioa kalkulatzeko balio duen funtzioa da. Puntuazio hori geroz eta altuagoa izan, egindako eskaerarekiko orduan eta garrantzia handiagoa izango du dokumentuak.

D dokumentua eta q_1, \dots, q_n hitzak dituen Q eskaera emanda, BM25 algoritmoak D dokumentuari ondorengo puntuazioa emango dio:

$$score(D, Q) = \sum_{i=1}^n IDF(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{avgdl}\right)} \quad (2.1)$$

non $f(q_i, D)$ funtzioak, q_i terminoaren maiztasuna D dokumentuan neurtzen duen, $|D|$ D dokumentuaren hitz kopurua den, eta $avgdl$ indizearen dokumentuen batz besteko luzera errepresentatzen duen. Bestalde, k_1 eta b aldagaiak libreak dira, eta 2.1 espresioa erabiltzen ari garen dokumentuen kolekziora egokitzea ahalbidetzen du. Nahiz eta balio hauek indizearen ezaugarrien arabera zehaztu behar diren, orokorrean $k_1 \in [1, 2, 2]$ eta $b = 0,75$ balioak esleitzen dira. Azkenik, $IDF(q_i)$ funtzioak q_i terminoaren dokumentuaren alderantzizko maiztasuna (ingelesezko *inverse document frequency*) neurtzen du.

BM25 algoritmoa hobeto ulertzeko, beharrezkoa da ekuazioaren bi funtsezko elementuak ondo ulertzea: $f(q_i, D)$ eta $IDF(q_i)$.

Esan bezala, $f(q_i, D)$ funtzioak q_i terminoaren D dokumentuan duen maiztasuna neurtzen

du, edo beste modu batera esanda, q_i hitza D dokumentuan zenbat aldiz azaltzen den zenbatzen du. Funtzio hau, ondorengo espresioaren bidez hobeto uler daiteke:

$$f(q_i, D) = \frac{q_i \text{ hitza } D \text{ dokumentuan azaltzen den aldi kopurua}}{D \text{ dokumentuaren hitz kopurua}} \quad (2.2)$$

Hori dela eta, eskaeraren hitz bat dokumentu batean askotan azaltzeak, dokumentu horrek garrantzia irabaztea ekarriko du, $f(q_i, D)$ funtzioak puntuazio altuago bat emango duelako.

Bestetik, $IDF(q_i)$ q_i terminoaren dokumentu batean duen alderantzizko maiztasuna bezala definitu dugu, baina definizio hau nahiko arrotza izan daiteke. Funtsean, IDF funtzioak, q_i hitzak indizea osoan duen garrantzia neurtzen du. Zenbait hitz, adibidez “eta”, “da” eta “bat” hitzak, askotan azaltzen dira dokumentu guztietan, baina ez dira garrantzitsuak.

Adibidez, “Etxera joan da” eskaera kontutan hartzen badugu, gerta liteke dokumentu baten “Etxera” eta “joan” hitzak ez izatea, baina “da” hitza dokumentuan zehar askotan azaltzea. Ondorioz, dokumentu horri puntuazio altua esleituko zaio, baina berez dokumentua ez da garrantzitsua egin den eskaerarentzat. Hori ekiditeko, 2.3 ekuazioak hitz bat dokumentu askotan azaltzen bada baina hau ez bada garrantzitsua, dokumentu horren azken puntuazioa baxuagoa izatea eragingo du. Beste modu batera esanda, IDF funtzioaren bidez, eskaeretan sor daitekeen zarata minimizatu egiten da.

$$IDF(q_i) = \log \frac{N - n(q_i) + 0,5}{n(q_i) + 0,5} \quad (2.3)$$

non N indizean dauden dokumentu kopurua eta $n(q_i)$ q_i hitza duten dokumentu kopurua den.

Azkenik, algoritmo honen hainbat aldaera desberdin sortu dira:

- BM11 eta BM15:

b aldagaiari muturreko balioak esleitzen zaizkionean BM25 funtzioa BM11 ($b = 1$) eta BM15 ($b = 0$) algoritmoetan bihurtzen da.

- BM25F:

BM25 funtzioaren aldaera bat da, non dokumentu bakoitza elementu desberdinez (adibidez estekak, testua eta titularra) konposatzen den eta elementu bakoitzak garrantzia maila desberdina duela kontsideratzen den.

- BM25+:

BM25 funtzioaren hobekuntza bat da BM25+. Izan ere, BM25 algoritmoarekin, eskaeraren hitza duten dokumentu luzeek, eskaeraren hitza ez duten dokumentu motzen puntuazio antzeko lor dezakete. Hau konpontzeko, BM25+ funtzioa δ aldagai aske berria (defektuz, $\delta = 1$ balioa ematen zaio) gehitzen da:

$$score(D, Q) = \sum_{i=1}^n IDF(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot (1 - b + b \cdot \frac{|D|}{avgdl})} + \delta \quad (2.4)$$

2.3 Web scraping teknika

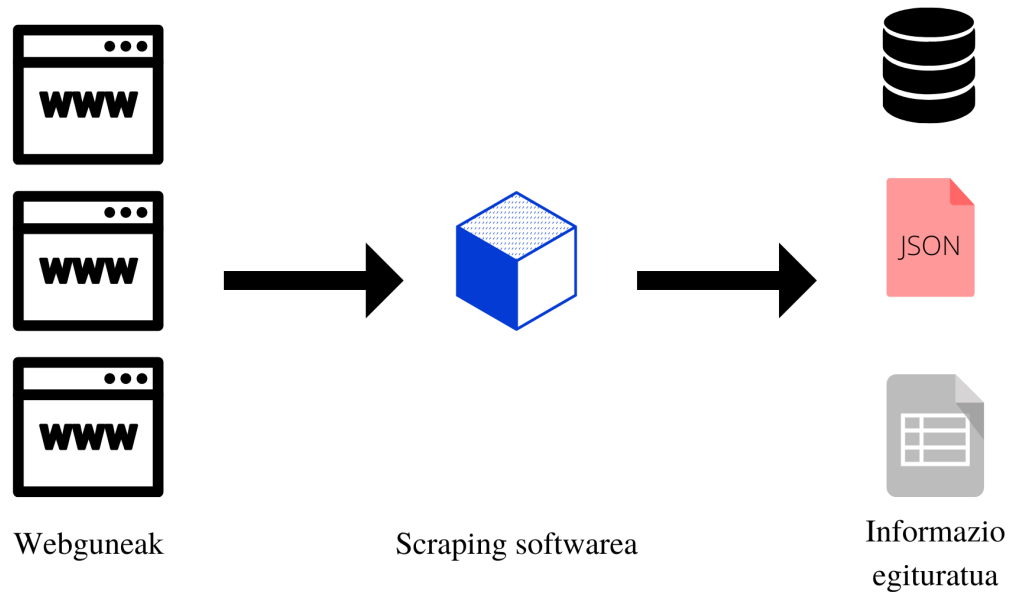
Web scraping teknika webguneetan egiturarik gabe dagoen testu, irudi eta beste zenbait elementuetatik informazio egituratua lortzean datza [Zhao, 2017]. Teknika hori eskuz egin badaiteke ere, orokorrean, scraping terminoa software baten bidez modu automatikoan webgune batetik informazioa eskuratzean erabiltzen da. Sarritan, lortzen den informazioa datubase batean gordetzen da, ondoren prozesatu eta analizatu ahal izateko. 2.5 Irudiak, web scraping teknikaren prozesu orokorra erakusten du. Prozesu horretan, *scraping* softwareak webguneen HTML kodetik informazioa hartu eta informazio hori formatu desberdinetako fitxategietan idazten du.

Teknika hori, bi zati desberdinetan bana daiteke: webgunearen bilaketa eta ondoren lortutako fitxategitik informazioa ateratzea. Webgunearen bilaketa, haren HTML edo XHTML fitxategia deskargatzean datza, eta hori erraz egin daiteke HTTP dei baten bidez. Behin kodea lortutakoan, edukiaren formatua analizatu egiten da, bertatik interesgarria den informazioa lortu ahal izateko.

Fitxategietatik informazioa ateratzeko, teknika desberdinak erabil daitezke eta webgunearen egituraren arabera zein erabiliko den erabaki behar da. Adibidez:

- Edukia eskuz kopiazea:

Webgune batetik informazio lortzeko modurik errazena, webguneko edukia hartu, kopiatu eta fitxategi batean itsatea da. Batzuetan webgunearen egituraren ondorioz, zaila izan daiteke automatikoki ataza hau egitea, hori dela eta, metodorik azkarrena eskuz egitea izan daiteke.



2.5 Irudia: Scraping prozesua.

- Testu patroiak parekatzea:

Webguneetako edukiek patroiak jarraitzen dituzte, hori dela eta, informazioa lortzeko programazio-lengoaiek eskaintzen dituzten espresio erregularrak erabiltzea oso tresna interesgarria da.

- HTML fitxategiak analizatzea:

Aurreko teknikarekin egiten den modu berean, HTML fitxategiek informazioa errepresentatzeko momentuan, patroia antzekoak erabiltzen dituzte. Orokorrean, kategoria berdineko datuak eredu berdina jarraitzen duten orrietan kodetzen dira, ondorioz, orri hauetatik informazioa ateratzea erraza da espresio erregularrak erabiliz.

- Ordenagailu bidezko ikusmena:

Azken urteetan, adimen artifizialaren bidez webguneetatik informazioa identifikatzeko eta ateratzeko esfortzuak egin dira [Zhao, 2017]. Teknika horren helburua, gizakiok webguneak interpretatzen ditugun modua simulatzea da.

Azkenik, gaur egun webguneetan dinamikoak diren datu asko aurkitu daitezke, adibidez, produktuen eta hauen prezioak, albistek, eguraldia etab. Web scraping teknika asko erabiltzen da mota hauetako datuak lortzeko.

2.4 Tokenizazioa, lematizazioa eta kategoria sintaktikoa

Hizkuntzaren prozesamenduko edozein ataza egiteko garaian, oso garrantzitsua da sarre-
ra bezala daukagun testua prozesatzea, esanguratsuagoa den forma batean jartzeko. Atal
honen helburua, lan honetan testua prozesatzeko erabili diren hiru tekniken (tokenizazioa,
lematizazioa eta kategoria sintaktikoa) oinarri teorikoa ematea da.

2.4.1 Tokenizazioa

Testu zati bat *token* deituriko unitate txikiagoetan banatzeko prozesua da tokenizazioa
[Chakravarthy, 2020]. Token bakoitza hitz bat, karaktere bat edo n-grama bat izatea da
ohikoena, baina tokenizatzeke modu asko existitzen dira, adibidez, esaldi tokenizazioa.

Datu bezala erabiltzen den testua tokenizatzea garrantzitsua da, alde batetik, errazagoa
delako lortutako token-ekin lan egitea eta horiek manipulatzeko, eta bestetik, orokorrean
token-ak testu hutsa baino interpretagarriagoak direlako. Gainera, gaur egun erabiltzen
diren hizkuntza prozesamenduko sare neuronalek, sarrera bezala token-ekin egiten dute
lan.

Bestalde, tokenizazioa hasiera batean sinplea izan daitekeen ataza bat dirudi, baina edo-
zein modutan tokenizatzeak arazoak sor ditzake. Horretaz gain, ez da posible hizkuntza
guztietarako balioko duen tokenizatzaile bat sortzea, hizkuntza bakoitzak bere ezauga-
rriak dituelako, eta horiek kontutan hartu behar direlako amaieran lortuko ditugun token-
ak esanguratsuak izatea nahi baldin badugu. Adibidez, ez da berdina frantsesa bezalako
hizkuntza bat tokenizatzea, non azentu-marka desberdinak kontutan hartu behar diren, in-
gelesa tokenizatzea, non oso ohikoak diren kontrakzioak ondo banatu behar diren, edo
txinera tokenizatzea, non erabiltzen den alfabetoa desberdina den.

Tokenizatzeke garaian, irizpide desberdin asko erabil daitezke. Esaterako, esaldi bat to-
kenizatzeke irizpide batzuk, hutsuneen arabera, puntuazio marken arabera edo hitzen ara-
bera banatzea izan daiteke [Chakravarthy, 2020]. Horietaz gain, gaur egun ere oso era-
biliak dira Treebank irizpidea, txioen tokenizatzailea eta *Byte-Pair Encoding* edo BPE
algoritmoa [Chakravarthy, 2020, Sennrich et al., 2015]. Ingelesaren tokenizazio irizpide
desberdinen adibideak 2.1 Taulan beha daitezke.

Irizpidea	Jatorrizko esaldia	Emaitza
Hutsune tokenizatzailea	"I was born in Tunisia in 1995."	["I", "was", "born", "in", "Tunisia", "in", "1995."]
Puntuazio marken tokenizatzailea	"It's true, Ms. Martha Jones!"	["It", "'", "s", " ", "Ms", ".", "Martha", "Jones", " !"]]
Txioen tokenizatzailea	"I'm doing good 😊"	["I'm", "doing", "good", " 😊"]]
Trebank tokenizatzailea	"Don't do that..."	["Do", "n't", "do", "that", "..."]]

2.1 Taula: Ingelesaren tokenizazio irizpide desberdinen adibideak.

2.4.2 Lematizazioa eta erro-bilaketa

Lematizazioa, flexionatuta edo deklinatuta dagoen hitz baten oinarriko lema lortzean datza [Khyani et al., 2021]. Prozesua zuzena izan dadin, beharrezkoa da sistemak hitzaren esanahia eta esaldian duen kategoria gramatikala ezagutzea, ondoren hitzaren lema lortu ahal izateko.

Lematizazioa hizkuntzaren prozesamenduko arlo desberdinetan egiten den prozesua da, batik bat, hitzen lemetatik informazio oso esanguratsua lor daitekeelako. Honen adibide, informazio erauzketa, sentimendu analisia edo bilaketa-motorrak dira.

Horretaz gain, lematizazioa erro-bilaketa edo *stemming*-arekin oso erlazionatuta dago. Erro-bilaketa erabiltzean, hitz bakoitzaren amaiera algoritmo heuristikoko baten bidez mozten da, hitzaren erroa lortu ahal izateko. Gainera, stemmer-ak, lematizatzaileak ez bezala, ez du kontutan hartzen hitzaren testuingurua esaldian, eta horregatik lematizatzaileak baino emaitza txarragoak ematen ditu orokorrean. Hala ere, stemmer bat sortzea errazagoa da, eta egin behar den atazaren arabera, lematizatzailearen antzeko emaitzak eman ditzake.

Horrela, bi tresna hauen helburua testu bateko hitzen oinarriko formak lortzea da. 2.2 Taulan lematizazioaren eta erro-bilaketaren arteko konparaketa ikus daiteke. “Hobetu” hitzaren kasuan, erro-bilaketak ez du emaitza zuzena eman. “Joaten” hitzarekin, berriz, bi teknikek emaitza bera eman dute.

Hitza	Lematizazioaren emaitza	Erro-bilaketaren emaitza
Hobetu	Ongi	Hobe
Joaten	Joan	Joan

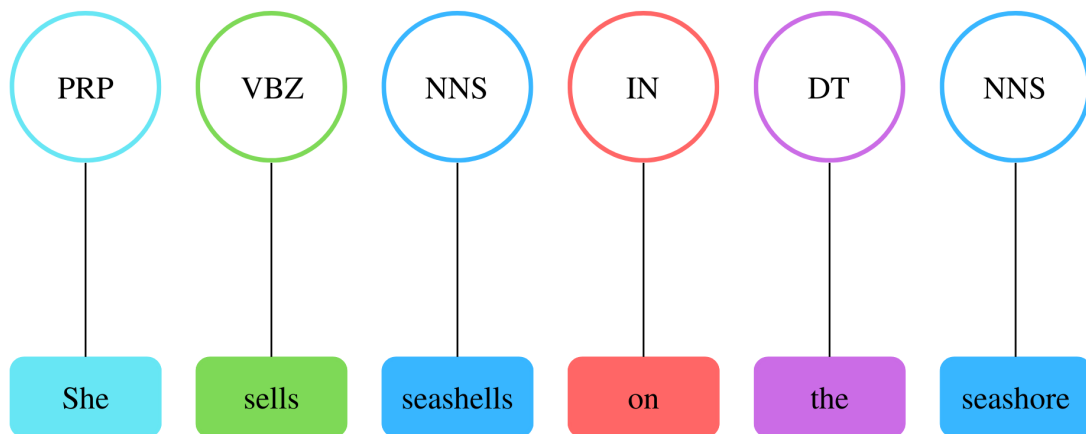
2.2 Taula: Lematizazioaren eta erro-bilaketaren konparaketa.

2.4.3 Kategoria gramatikalen etiketatzea

Kategoria gramatikalen etiketatzea (ingelesezko *part-of-speech tagging* edo *POS tagging*), testu bateko hitz bakoitzari esaldian dagokion kategoria gramatikala esleitzaren prozesuari deitzen zaio [Márquez et al., 2000]. Ingelesezko esaldi baten adibide bat ikus daiteke 2.6 Irudian. Adibidean, Penn Treebank irizpidea jarriatuz esaldiaren hitz bakoitzari etiketa bat esleitu zaio [Santorini, 1990]. Gainera, etiketa bakoitza kolore desberdin batekin errepresentatu da.

Hitzen kategoria gramatikalak jakiteak, oso baliagarria izan daitekeen informazioa ematen du. Esate baterako:

- Hainbat hizkuntzetan hitz berdinak kategoria gramatikal desberdina izan dezake, adibidez, euskaran “hori” hitza izen arrunta, adjektiboa, aditza eta determinante erakuslea izan daiteke.
- Esaldi baten sintaxia errazago ezagutzeko. Adibidez, kategoria gramatikala jakinda, errazagoa izan daiteke esaldiaren subjektua zein den jakitea.
- Termino bat baino gehiagoko hitzak identifikatzeko. Kategoria gramatikalaren bidez, terminoek jarraitzen dituzten patroiak bilatu daitezke, horrela termino anitzeko hitz berriak ezagutzeko.
- Informazio erauzketaren atazan dokumentuak filtratzeko. Orokorrean, dokumentu bat errepresentatzeko K termino esanguratsuenak izenak izan ohi dira.



2.6 Irudia: Ingelesezko esaldi baten kategoria gramatikalen etiketatzea.

3. KAPITULUA

Garatutako implementazioa

Lan honetan sortu den chatbot-aren atzean dagoen implementazioa ulertu ahal izateko, beharrezkoa da lehenengo chatbot-ak eskaintzen dituen bi funtzionaltasun nagusiak aztertzea.

Juliet chatbot-ak bi funtzionaltasun nagusi ditu: galdera irekien albisteak bilatzea eta kategoria baten berriak ematea. Bi kasuetan, erabiltzaileak aldi berean mezu gehiegi jaso ez ditzan, egiten den eskaera bakoitzeko (galdera irekia ala kategoria) gehienez hiru albiste itzultzea erabaki da. Erabiltzaileak galdera ireki bat egiten dionean Juliet-i, chatbot-ak eskaereriko garrantzitsuak diren albisteak itzuliko dizkio erabiltzaileari. Bestetik, Berriaren webgunean ¹ albisteak kategoria desberdinetan banatuta azaltzen dira, adibidez, kirolak, kultura, bizigiro... Hauek bilaketak egiteko baliagarriak izan daitezke, izan ere, agian erabiltzaile batek kategoria konkretu bati buruzko albisteak baino ez ditu jakin nahi. Hori dela eta, erabiltzaileak kategoria baten izena eskaera moduan jartzen badu, chatbot-ak kategoria horren lehen hiru albisteak bidaliko dizkio.

Horretaz gain, erabiltzaileak dialogo-sistema nola erabiltzen den ez badaki, “Laguntza” idatziz, chatbot-ak bilaketa egiteko argibideak dituen mezu bat bidaliko dio.

Azkenik, eskaera bat (galdera irekia edo kategoria) egin ostean, aurkitutako albisteak zenbakituta eta hauek aukeratzeko botoiak jasotzen ditu erabiltzaileak. Modu horretan, erabiltzaileak zenbaki horiek erabil ditzake albiste bati buruz gehiago jakiteko:

- 1, 2 edo 3 zenbakiak idatzi.

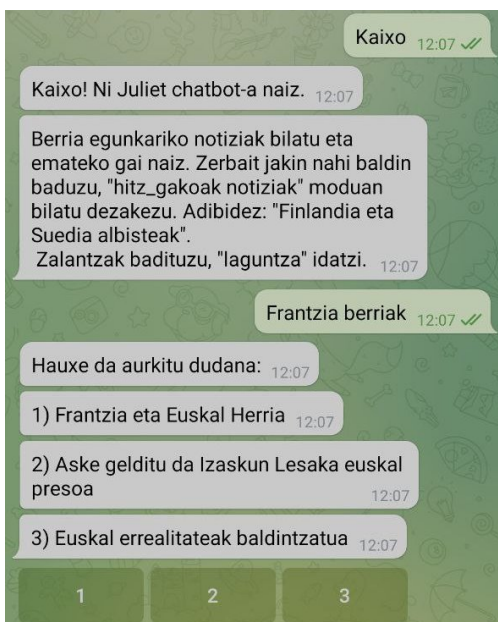
¹<https://www.berria.eus/>

- “Bat”, “Bi” edo “Hiru” hitzak idatzi.
- “Lehenengoa”, “Bigarrena” edo “Hirugarrena” hitzak idatzi.
- Bistaratu diren botoiak sakatu.

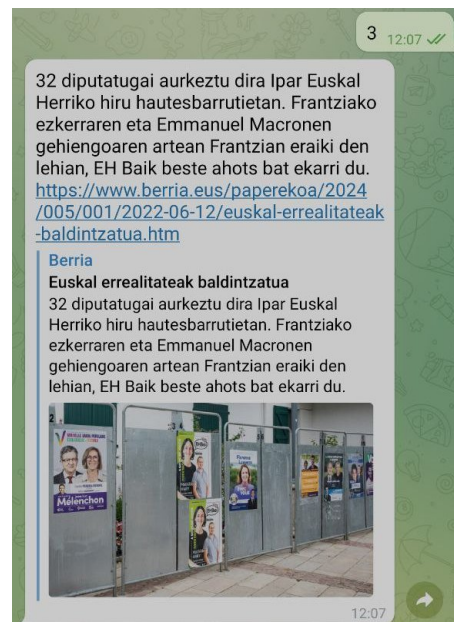
Hori egitean, chatbot-ak zenbakia jasoko du, eta zenbaki horri dagokion albistearen edukia (defektuz azpigitulua, hau ez balego, albistearen lehen paragrafoa) eta esteka mezu bakar batean bidaliko dizkio erabiltzaileari.

Bi funtzionaltasunak erakusten dituzten elkarrizketak 3.1 eta 3.2 Irudietan ikus daitezke. Lehenengoa irudian, erabiltzaileak Frantziari buruz galdetzen du, hau da, galdera ireki bat egiten du eta chatbot-ak aurkitutako albisteak itzultzen dizkio. Bigarrengoan, berriz, erabiltzaileak kategoria bateko albisteak eskatzen ditu, eta dialogo-sistemak kategoria horren azken hiru berriak ematen dizkio. Bi kasuetan, erabiltzaileak interesgarria iruditzen zaio albistearen zenbakia idatzi du, eta chatbot-ak haren edukia eta esteka itzuli dizkio.

Amaitzeko, dialogo-sistema honek ematen dituen albiste guztiak, Berria egunkariaren webgunetik hartzen dituela kontuan hartu behar da. Hori dela eta, erabiltzaile batek Berrian egun horretan publikatu ez den artikulu bati buruz galdetzen badu, chatbot-ak ez du ezer aurkituko.

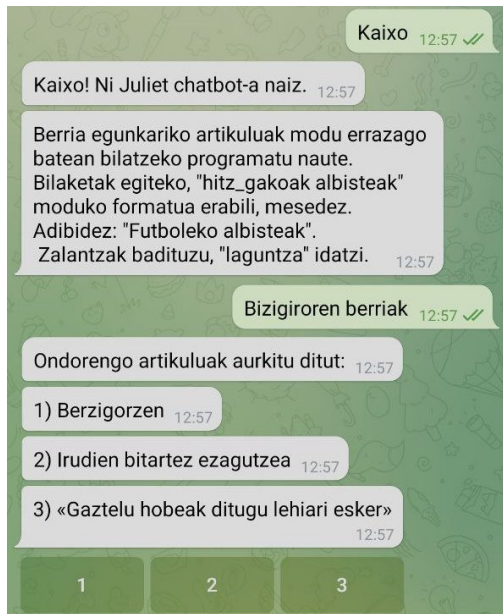


(a) Albiste baten eskaera.

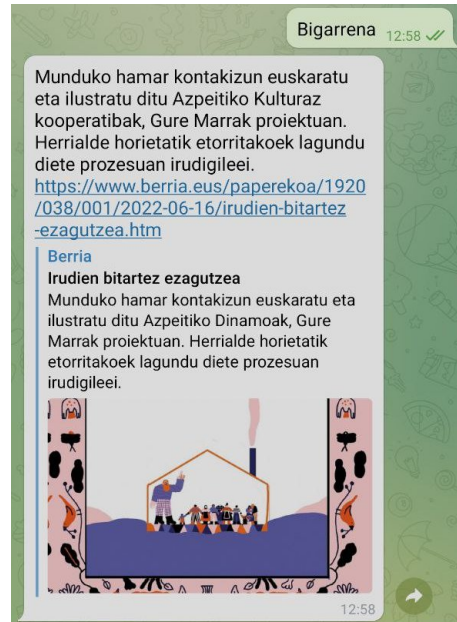


(b) Albistearen edukia eskuratzea.

3.1 Irudia: Albiste bat eskuratzeko elkarrizketa eredua.



(a) Kategoría bateko albisteak eskuratzea.



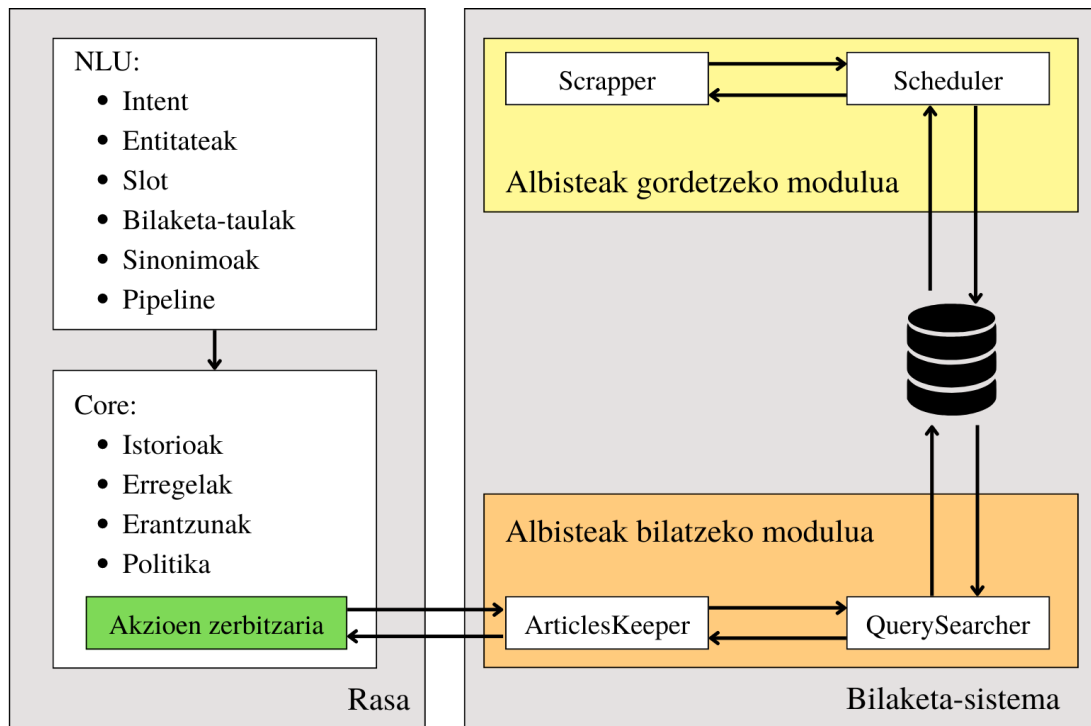
(b) Albistearen edukia eskuratzea.

3.2 Irudia: Kategoría bateko albisteak eskuratzeko elkarriketa eredua.

3.1 Sistemaren arkitektura

Software ingenieritzaren praktika onak mantentzeko asmoz, sortutako dialogo-sistema modulo desberdinen bidez eraiki da. Horrela, egingako implementazioa hiru atal nagusitan banatzen da: Rasa-ren bidez sortutako chatbot-a, Berria-ko egunkaritik *scraping* bidez albisteak lortu eta indizean gordetzen dituen modulua, eta lengoai-naturala erabiliz indizeari deiak egitea ahalbidetzen duen modulua. Banaketa hori eginez, hiru atalak beraien artean komunikatzen dira erabiltzaileen eskaerak zerbitzatzeko, baina modu independentean garatu daitezke. Gainera, albisteak lortzeko eta albisteak bilatzeko moduluen konbinaketari, bilaketa-sistema deitu zaio.

3.3 Irudian, ezkerreko aldean Rasa-rekin erlazionatuta dauden kontzeptuak ikusi daitezke. Eskubiko aldean, berriz, bilaketa-sistemarekin zerikusia daukaten modulu eta klaseak agertzen dira. Irudian azaltzen diren geziek atal desberdinen arteko elkarrekintzak errepresentatzen dituzte.



3.3 Irudia: Juliet chatbot-aren arkitekturaren eskema orokorra.

3.1.1 Rasa

Rasa *framework*-a erabiltzean, defektuz dialogo-sistemaren inplementazioa atal desberdinetan banatzen da. Horrela, sortutako sistemaren atzean dagoen oinarria ulertzeko, beharrezkoa da atal desberdinek nola funtzionatzen duten, horien helburua zein den eta atal desberdinen arteko elkarrekintza nolakoa den ikustea.

Lehenik eta behin, erabiltzailearen eta chatbot-aren arteko elkarrizketak nolakoak izango diren zehaztu behar da, elkarrizketa horietatik istorio eta erregela desberdinak idatzi ahal izateko. Gainera, erabiltzaileen eta dialogo sistemen elkarrekintzak hasieran definitzeak, inplementaziorako beharko diren intent, akzio, erantzun eta entitate kopuruen ideia bat lortuko da.

Horrela, dialogo-sistemaren bi funtzionaltasun nagusiak betetzeko, ondorengo istorioa definitu da: Lehenengo, erabiltzaileak dialogo-sistema agurtuko du. Ondoren, chatbot-ak bueltan agurtuko dio, eta bilaketak egiteko azalpenak emango dizkio. Horren ostean, erabiltzaileak albiste edo kategoria baten notizien eskaera egingo du, eta sistemak bilaketa-aren bidez lortutako emaitzak itzuliko dizkio. Bistaratutako berriren bat erabiltzailearentzat interesgarria bada, notziaren zenbakia idatziko du, honi buruz informazio gehiago nahi

duela adierazteko, eta chatbot-ak eskatutako notiziaren edukia eta esteka bidaliko dizkio, mezu bakar batean. Azkenik, nahiz eta “elkarrizketa” modu sekuentzian definitzen den, ez da beharrezkoa pauso guztiak betetzea beti, hau da, istorioak elkarrizketaren patroia orokorra erreprezentatzen du, baina horrek ez du esan nahi beti atal guztiak errepikatu behar direnik. Adibidez, albiste baten eskaera egiteko, ez da beharrezkoa alde aurretik chatbota agurtzea, eskaera egitea berriak jasotzeko nahikoa izan beharko litzateke. Elkarrekintza hori definitzen duen istorioa erakusten du 3.4 irudiak. Bertan, erabiltzailearen intent-ak gorritz, dialogo-sistemaren erantzunak berdez eta akzioak urdinez ikusi daitezke.

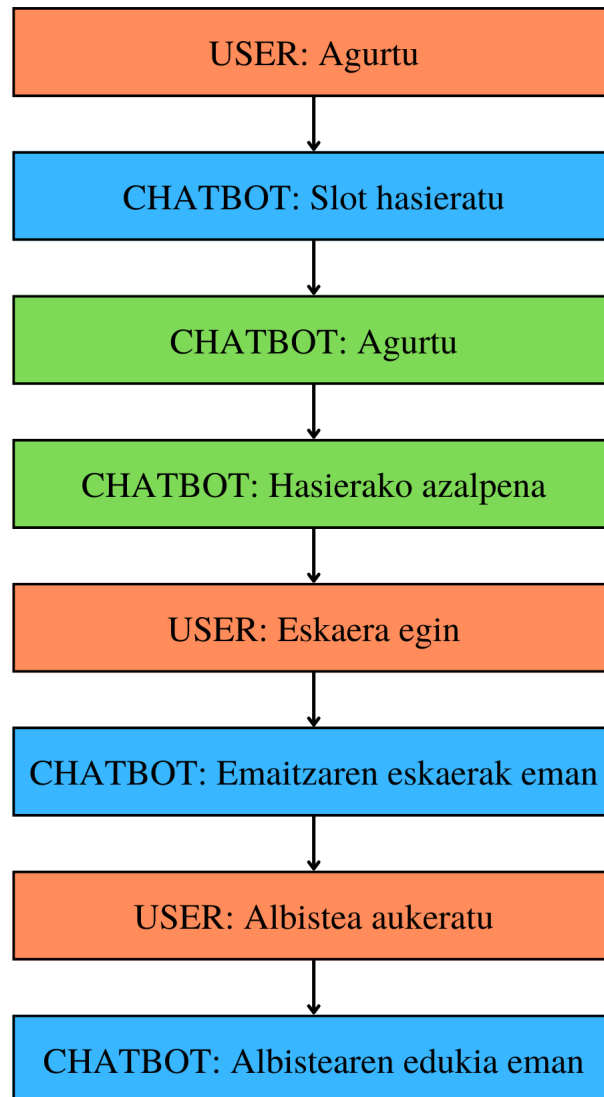
Erabiltzaileek chatbot-aren funtzionamenduari buruz zalantzak badituzte, “Laguntza” hitza idazten badute, sistemak laguntza mezu bat bistaratuko du. Hori egin ahal izateko, beste erregela bat idatzi da. Laguntza mezua erakusteko erregela erreprezentatzen du 3.5 irudiak.

Definitu den istorioari eta erregelari erreparatuz, argi ikus daiteke lau intent desberdin erabili behar direla, guk nahi ditugun elkarrizketak sortu ahal izateko. Horrela, sortu diren intent-ak ondorengoak dira: dialogo-sistema agurtzeko, berri baten edo kategoria baten notiziak eskatzeko, albiste baten edukia eskatzeko eta laguntza eskatzeko intent-ak, hain zuzen ere.

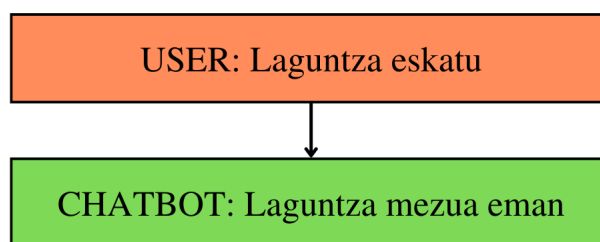
Erabiltzaileen mezuetatik, soilik kategoria (kasu guztietan ez dena lortzen) eta artikuluzenbakia interesatzen zaizkigunez, bi horiek lortzeko bina entitate eta *slot* definitu dira. Horrela, chatbot-aren modeloak mezuetatik entitateak lortuko ditu, eta entitateen balioak *slot*-etan gordeko dira, ondoren hauek erraztasunez atzitu ahal izateko. Bestetik, erabiltzaileak albisteen eskaera bat egiten duen bakoitzean, lortutako emaitzak bilaketa-moduluaren `ArticlesKeeper` klasean gordetzen dira, ondoren lortutako albiste horien edukia modu erangikorrago batean itzuli ahal izateko. Ondorioz, klasearen egoera eguneratuta izan behar da uneoro, eta hori egiteko *slot* batean gordetzen da JSON formatuarekin.

Inplementazioan definitu diren bi entitateek (albisteen kategoriak eta albisteen zenbakiak), balio kopuru mugatu bat daukate, izan ere, egunkariaren kategoria kopurua eta chatbot-ak ematen dituen albiste kopuruak finituak dira. Hori dela eta, entitate horiek erabiltzaileen mezuetatik ateratzeko, bi bilaketa-taula (*lookup-table*) definitu dira.

Bestalde, entitateak deklinatuta azaldu daitezkeenez, entitateen berezko balioak lortzeko sinonimoak erabili dira, adibidez “Kirolen berriak”. Horretaz gain, albisteen zenbakia aukeratzeko modu desberdinak daudenez, aukera bakoitzaren balio numerikoa ere lortzeko sinonimoak erabili dira. Esate baterako, “1”, “Bat” eta “Lehenengoa” hitzek albiste berdinari egiten diote erreferentzia, hori dela eta, chatbot-ak mezu hauetako bat jasotzen



3.4 Irudia: Erabiltzailearen eta chatbot-aren arteko elkarrekintza.



3.5 Irudia: Laguntza mezua erakusteko erregela.

duenean, albistearen zenbakiaren *slot*-ean 1 balioa gordeko du.

Agurtzeko, laguntzeko eta errore emateko mezuak erantzun (*responses*) moduan definitu dira, testu hutsezko mezuak baitira. Akzio pertsonalizatuei dagokionez, *ArticlesKeeper* hasieratzeko, berrien titularrak bilatzeko eta albiste baten edukia itzultzeko akzio bana idatzi dira.

Azkenik, proiektuaren tamaina eta entrenamendurako erabili den datu kopuru murrizta kontuan hartuta, defektuzko *pipeline*-a eta politikak erabiltzea erabaki da. *Pipeline*-aren aldetik, DIET (*Dual Intent and Entity Transformer*) sailkatzailea da elementurik garrantzitsuenak [Bunk et al., 2020]. DIET sailkatzaileak *transformer* sare neuronalak erabiltzen ditu intent-ak eta entitateak sailkatzeko, eta aldeaz aurretik entrenatutako modelo handirik erabili gabe, bi ataza hauetan oso emaitza onak lortzen ditu. Politikari dagokionez, TED (*Transformer Embedding Dialogue*) politika azpimarratu behar da [Vlasov et al., 2019]. TED politikak, chatbot-ak pauso bakoitzean egin behar duen ekintza aukeratzeko, elkarriketan esanguratsuak izan diren mezuak hartzen ditu kontutan. Modu horretan, dialogo-sistemak espero ez duen mezurik jasotzen badu, horrek ez du eragin negatiborik sortzen hurrengo ekintzak iragartzeko garaian.

3.1.2 Bilaketa-sistema

Bilaketa-sistemaren helburua Berria egunkariko albisteekin indizeak eguneratzea eta *Rasa*-ren akzioen fitxategitik indizera lengoai naturala erabiliz deiak egitea da, erabiltzaileek egindako eskaerak ahalik eta modu azkar eta eraginkorrean zerbitzatuz. Bilaketa-sistema modulu eta klase desberdinetan banatuta dago: *scraper*, *scheduler*, *QuerySearcher* eta *ArticlesKeeper*.

Berria egunkaririk notiziak *scraping* teknikaren bidez lortzeko funtzioak *scrapper* modulan daude. Bertan egiten dira egunkariaren HTML fitxategien eskuraketa, informazioaren erauzketa eta datuen aurreprozesaketa. Horretaz gain ere, erabilitako lematizazioa irteera moduan NAF fitxategi bat sortzen du, eta sortutako fitxategi horietatik hitzen lema eta kategoria gramatikalak lortzeko metodoak ere *scrapper* modulan aurki daitezke.

Bestalde, planifikazio-modulua (*scheduler*), denbora guztian exekutatzen egoteko eta egunean behin funtzio konkretu bat exekutatzeke pentsatuta dago. Funtzio horrek, Berria egunkariko albiste guztiak hartu, eta horiekin dagozkien indizeak eguneratzen ditu.

ArticlesKeeper klasearen helburua Rasa-ren akzioen zerbitzariaren eta *QuerySearcher* klasearen arteko bitartekari lanak egitea da. Horretaz gain, bilaketan ostean lortutako emaitzak ere gordetzen ditu, ondoren berrien edukia azkarrago itzuli ahal izateko.

Amaitzeko, *QuerySearcher* klasea bilaketak egiteko klasea da. *ArticlesKeeper* klasetik bidalitako eskaerak, modulu honetan prozesatu eta ondoren dagokion indizean bilaketa prozesua egiten da. Azkenik, lortutako emaitzak *ArticlesKeeper* klaseari itzultzen zaizkio.

3.2 Funtzionalitateak eta hauen inplementazioa

Sortutako dialogo-sistemaren funtzionalitate nagusiak, erabiltzaileak egindako eskaerarekiko garrantzia handiena duten hiru artikulu itzultzea eta kategoria konkretu bateko azken hiru berriak ematea dira. Atal honetan, bi funtzionaltasun horiek eskaintzeko beharrezkoak diren prozesuak eta atzetik ematen diren pauso guztiak azaltzen dira.

3.2.1 Berriak lortzea

Lehenik eta behin, beranduago chatbot-ak erakutsiko dituen notizien izenburuak, esteka eta edukia eskuratu behar dira. Horretarako, lehenik eduki hauek *scraping* eginez eskuratu ditugu, ondoren prozesatu egingo dira, eta amaieran indizean gordeko dira.

Esan bezala, lehen pausoa artikuluen edukiak lortzea da. Hori egiteko, beharrezkoa da Berria egunkariaren webgunearen egitura ezagutzea. Kasu honetan, artikulu guztiak lortu nahi ditugunez, notizi bakoitzaren edukiak lortzeko, beharrezkoa da bi HTTP dei egitea: lehenengoarekin, kategoria baten (adibidez, kirolak, kultura...) HTML kodea lortuko da

eta hortik, bertan dauden artikuluko guztien izenburua eta esteka aterako dira, eta ondoren, artikuluko bakoitzaren esteka erabiliz, honen HTML fitxategitik berriaren azpititulua (hau existituko balitz) eta lehen paragrafoa eskuratuko dira.

Horren ostean, lortutako artikuluko bakoitzarentzat, titular, azpititular eta lehen paragrafoarekin karaktere-kate bakar bat osatuko dugu. Karaktere-kate horren letra larri guztiak, letra xehe bihurtu, eta ondoren, karaktere-katea tokenizatu eta lematizatu egingo da. Prozesu hori egitea oso garrantzitsua da, ondoren bilaketa guztiak artikuluko bakoitzaren prozesatutako karaktere-kate horren aurka egingo direlako.

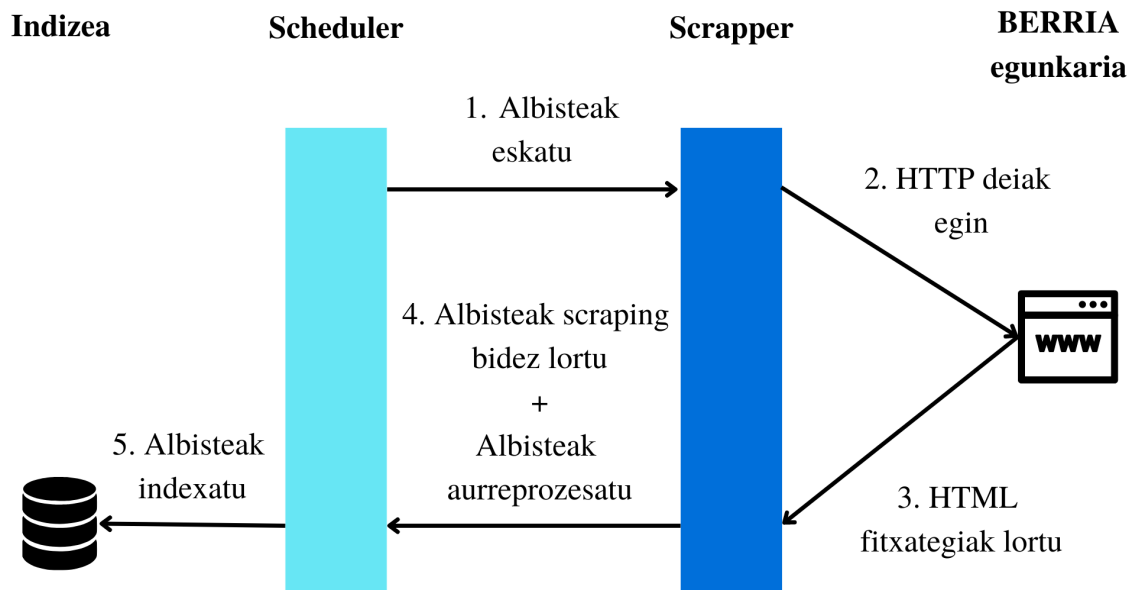
Berri guztiak lortu eta prozesatu ditugunean, indexatu egin behar dira. Horrela, lortutako artikuluko bakoitzarentzat dokumentu bat sortu eta dagokion indizean gordeko dugu. Dokumentu horrek lau atal izango ditu: jatorrizko titularrak (aurreprozesaketarik gabekoa), notiziaren esteka, berriaren azpititulua edo lehen paragrafoa eta bilaketarako sortu den karaktere katea. Lehenengo hiru atalek, ez dute eraginik bilaketan, baina garrantzitsua da albistearen jatorrizko informazioarekin gordetzea, bilaketetatik lortzen diren emaitzak zuzenak izan daitezten.

Azkenik, webgunean albisteak kategoriatan banatuta daudenez, indexatzerako garaian banaketa hori mantentzea erabaki da. Albiste bakoitzaren dokumentua horri dagokion kategoriaren indizean gordeko da, ondorioz, kategoria bakoitzak indize bat izango du. Hori erabilgarria izango da kategoria baten albisteak lortzeko. Kategorien indizeez gain, albiste guztiak gordeko dituen beste indize bat sortuko da, hau da, azken hori kategoria guztien indizeen konbinaketa bat da. Bestalde, berriak eskuratzeko prozesua egiten den bakoitzeko, indize guztiak hustu egiten dira, eta lortutako albiste berriekin eguneratzen dira, hori dela eta, ezin dira egun desberdinetako notiziak lortu, hauek ez baitaude indizean gordeta.

3.6 Irudiak berriak lortzeko prozesu osoaren laburpena errepresentatzen du. Funtsean, scheduler moduluak scrapper moduluari albisteen eskaerak egiten dizkio. Segidan, scrapper moduluak albisteen HTML fitxategiak lortu, fitxategi horietatik albisteak atera eta albisteak aurreprozesatzen ditu. Amaitzeko, prozesatutako albisteak scheduler-era itzultzen eta indizean gordetzen dira.

3.2.2 Albiste bat bilatzea

Erabiltzaileak gai konkretu baten berriak lortu nahi dituzenean, “Futboleko berriak” moduko eskaera bat egingo du. Horrela, Rasa-k emandako mezuaren intent-a albiste baten



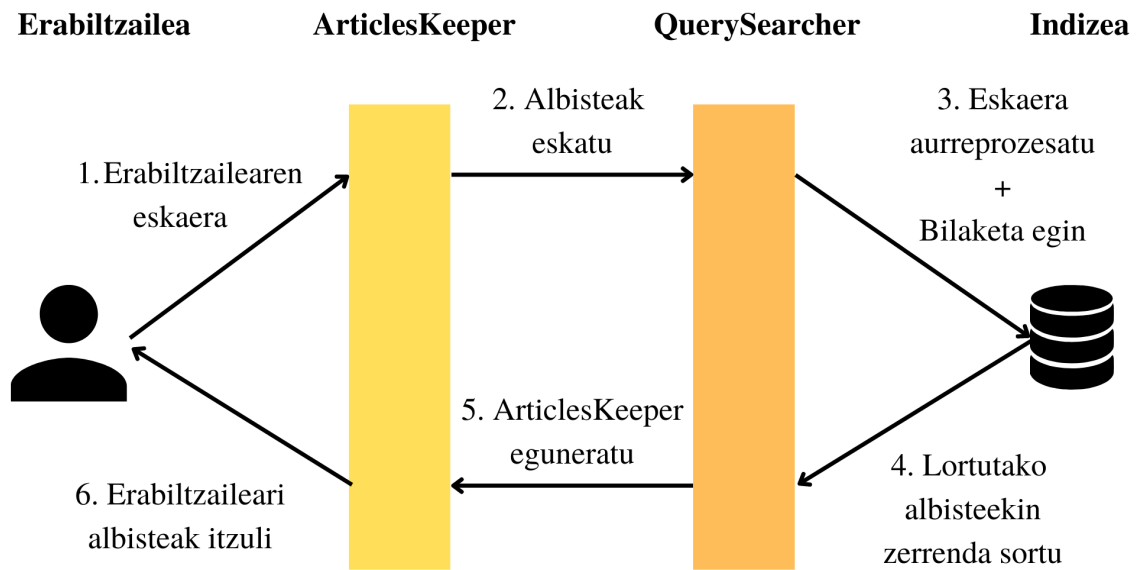
3.6 Irudia: Berriak lortzeko prozesua.

eskaera dela detektatuko du, erabiltzailearen mezua Rasa-ko akzioen zerbitzarira iritsiko da eta bertatik bilaketa-prozesua martxan jarriko da.

Kasu horretan, erabiltzailearen mezutik ez denez kategoriarik lortu, bilaketa artikuluz guztien aurka egingo da. Horrela, bilaketa egiteko lehen pausoa, erabiltzailearen mezu osoa tokenizatzea eta lematizatzea izango da. Tokenizatze eta lematizatze erabili den tresnak, sarrera lematizatzeaz gain, hitz bakoitzaren kategoria gramatikala ere lortzen duenez, izenak eta adjektiboak ez diren beste hitz guztiak eskaeratik ezabatu egingo dira. Horretaz gain, eskaerak egiteko garaian “notiziak” edo “albisteak” bezalako hitzak erabili behar dira intent sailkapena zehatzagoa izateko. Hitz horiek bilaketaren katean mantentzeak emaitzetan eragin negatiboa sor dezaketenez, eskaeratik ezabatu egiten dira. Horrela, “Futboleko berriak” eskaera emanda, aurreprozesatzea egin ostean, “Futbol” katearekin geldituko gara. Ondorioz, iragazketa-metodo horren bidez, erabiltzaileak bilatu nahi duen kontzeptuarekin gelditzea lortzen dugu.

Horren ostean, prozesatutako emaitza artikuluz guztiak dituen indizearen aurka bilatuko da. Horrela, bilaketa egitean, erabilitako liburutegiak dokumentu bat kontutan hartzeko, gutxienez eskaeraren terminoren bat dokumentuaren bilaketa-katean azaldu beharko da.

Ondoren, lortutako dokumentuen artean, BM25F algoritmoa erabiltzen da dokumentuak puntuatzeko. Puntuazioa, dokumentuen sailkatzearekin egiten den bezala, dokumentu bakoitzaren bilaketa-katearen aurka egiten da. Modu horretan, bilaketa prozesuak puntuazio



3.7 Irudia: Albiste bat bilatzeko prozesua.

altuena jasotzen duten dokumentuak itzuliko ditu. Gehienez hiru dokumentu itzuliko dira, baina gerta liteke egindako eskaerarekiko dokumenturik ez aurkitzea edo hiru dokumentu baino gutxiago aurkitzea. Bestalde, kontutan izan behar da dokumentu bakoitza titularra, esteka, edukia eta bilaketa katea gordetzen dituen objektu bat dela.

Azkenik, lortutako dokumentuak ArticlesKeeper objektuan gordeko dira, beranduago erabiltzaileak albisteren baten edukia jaso nahiko balu, azkarrago jaso dezan. Horren ostean, dokumentuak akzioen zerbitzarira itzuliko dira, dokumentu bakoitzaren titularra hartuko da eta Telegram-ek eskaintzen duen API-a erabiliz, erabiltzaileari aurkitutako notizien titularrak erakutsiko zaizkio. Kasu horretan, artikuluen aukeratzea idatziz egin badaiteke ere, aukeraketa errazteko titular bakoitzeko botoi bat bistaritzen da.

3.7 Irudiak albiste baten bilaketa-prozesua laburtzen du. Erabiltzaileak eskaera bat egin, eskaera hori ArticlesKeeper-era iritsi eta QuerySearcher klasera bidaliko da. Bertan, eskaera aurreprozesatu eta dagokion indizearen aurka bilaketa egin ostean, bilaketatik lortutako emaitzak ArticlesKeeper-era itzuli eta bertan gordeko dira. Azkenik, lortutako albisteak erabiltzaileari bidaltzen dira.

3.2.3 Kategoria bateko albisteak lortzea

Erabiltzaileak kategoria baten albisteak lortu nahi dituzenean, “Ekonomiaren berriak” moduko eskaera bat egingo du. Modu horretan, Rasa-k mezuaren intent-a albisteen eskaera dela detektatzeaz gain, “Ekonomiaren” hitza kategoriaren entitate bezala sailkatuko du, eta hori aldi berean, “Ekonomia” sinonimoari lotuko du. Horrela, akzioen zerbitzari-tik kategoriaren *slot*-aren balioa hartzean, “Ekonomia” hitza hartuko du, “Ekonomiaren” beharrez.

Horren ostean, akzioen zerbitzarian kategoriaren entitatearen balioa hartuko da. Lortutako kategoria zuzena bada, kategoria hortako albisteak ekarriko dira. Bestela, modeloak entitate bezala identifikatu duen balioarekin indizek ez bada existitzen, hau da, modeloak erabiltzailearen mezutik kategoria lortzean huts egin badu, orduan erabiltzailearen mezu guztiarekin lehenago azaldu den albisteen bilaketa normala egingo da.

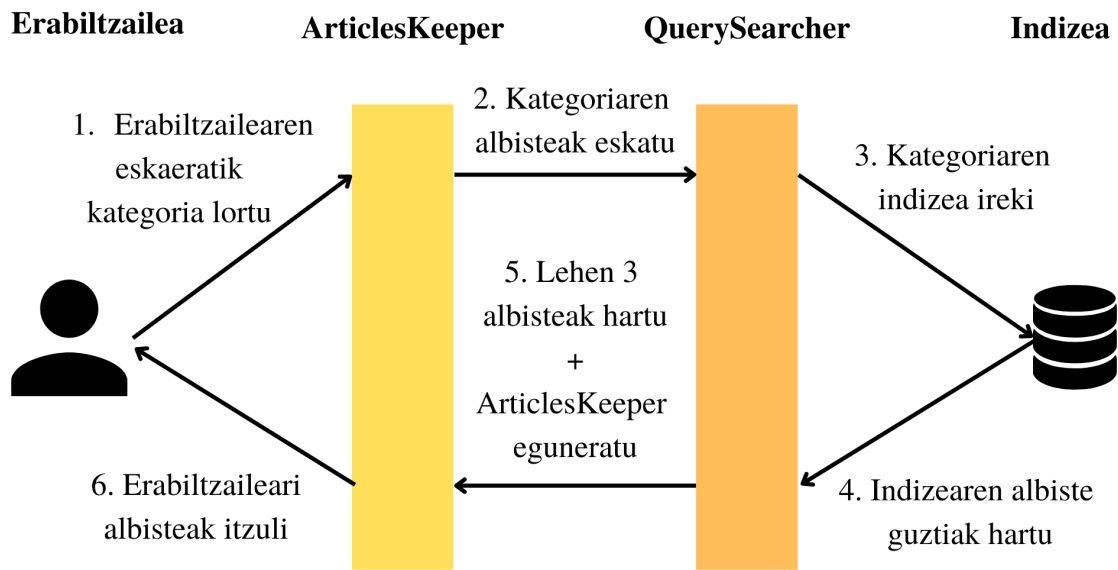
Kategoria baten albisteak lortzeko, kategori konkretu hortako indizea ireki eta bertatik lehenengo hiru dokumentuak hartuko dira. Ondoren, bilaketa normalarekin egiten den modu berean, hiru dokumentu horiek `ArticlesKeeper` objektuan gordeko dira, eta akzioen zerbitzariari pasako zaizkio.

Azkenik, akzioen zerbitzariak dokumentuak lortzen dituzenean, dokumentuetatik titularrak lortu, eta horiek Telegram-en API-aren bidez erabiltzaileari erakutsiko dizkio. Hontaz gain, titular bakoitzeko botoi bat bistaratuko du, aukeraketa errazteko.

3.8 Irudiak kategoria bateko albisteak lortzearen prozesua erakusten du. Lehenik eta behin, erabiltzailearen mezutik albisteen kategoria lortzen da eta hori `QuerySearcher` klasera bidaltzen da. Ondoren, kategori horren indizeko albiste guztiak hartu eta `ArticlesKeeper`-era bidaltzen dira. Amaitzeko, klase horrek lehen hiru albisteak bere memorian gorde eta erabiltzaileari bidaltzen dizkio.

3.2.4 Albiste baten edukia lortzea

Albiste baten edukia lortzeko, erabiltzaileak lehenik eta behin bilaketa bat egin behar du. Ondoren, bilaketatik lortutako artikulu bakoitzari zenbaki bat esleitzen zaio, eta zenbaki horien bitartez egin daiteke albistearen aukeraketa. Esate baterako, lehenengo notiziaren edukia lortzeko, “bat”, “lehenengo”, “lehenengoa”, “1” bezalako hitzak erabiltzeaz gain, 1 zenbakia duen botoia ere sakatu daiteke. Horrela, erabiltzaileak notizia aukeratzeko mezu idazten duenean, egindako aukeraketa entitate moduan identifikatzen da, eta ondoren



3.8 Irudia: Kategoriaren albisteak lortzeko prozesua.

sinonimoak erabiliz aukeraketa horren balio numerikoa lortzen da. Adibidez, “lehenengo” idatziz gero, 1 balioa lortuko da, “bigarrena” idatziz gero, 2 balioa etab. Azkenik, lortutako balio numerikoa *slot* batean gordetzen da.

Kontuan izan behar da, edozein motatako bilaketa bat egiten den bakoitzean, lortutako dokumentuak *ArticlesKeeper* objektuan gordetzen direla, zerrenda moduan. Horrela, akzioen zerbitzaritik aukeratutako notiziaren indizearen *slot*-a irakurtzen da, eta lortutako balioarekin, *ArticlesKeeper* objektutik notiziaren edukia eta esteka eskuratzen dira. Ondorioz, erabiltzailearen mezutik notiziaren zenbakia lortu denean, nahiko da emaitzen zerrenda hontako indizean dagoen dokumentua hartzea, eta bertatik notiziaren esteka eta edukia akzioen zerbitzariari pasatzea.

Behin aukeratutako notiziaren edukia eta esteka lortu direnean, horiek kateatuz mezu bat sortu eta erabiltzaileari bidaliko zaio Telegram-en API-a erabiliz. Notiziaren edukia luzee-gia balitz (Telegram-ek mezuen luzera maximo bat du), hau moztu egiten da, eta ondoren esteka gehitzen zaio.

3.9 Irudiak albiste baten edukia lortzeko prozesuaren laburpena erakusten du. Lehenengo erabiltzailearen mezutik albistearen zenbakia lortu behar da. Horren ostean, *ArticlesKeeper*-etik zenbaki horri dagokion albistea hartu, mezua prestatu eta erabiltzaileari bidaltzen zaio.

4. KAPITULUA

Ebaluazioa eta lortutako emaitzak

Atal honen helburua, chatbot-a ebaluatzeko erabili diren ebaluazio protokoloak azaltzea eta lortutako emaitzak aztertzea da.

Dialogo-sistema baten ebaluazioa, ataza konplexua izan daiteke, chatbot-ak sortzen dituen elkarrizketak zuzenak ala okerrak diren esatea anbigua izan daitekeelako. Izan ere, kontzeptu berdina adierazteko modu desberdinak erabil daitezke: kasu bakoitzean esaldi bat baino gehiago zuzenak izan daitezke eta horrek metrika automatikoak erabiltzea aukera ona ez izatea eragiten du.

Hori guztia kontutan hartuta, ebaluazioa giza-erabiltzaileekin egitea erabaki da. Horretarako, ebaluazio-protokolo bat definitu da, ondoren zortzi pertsonetako lagin batekin protokoloa jarraitu da, eta azkenik, lortutako emaitzen analisi kuantitatibo eta kualitatiboa egin da.

4.1 Ebaluazio-protokoloa

Ebaluaziotik lortzen diren emaitzak esanguratsuak eta homogeneoak izan behar direnez, beharrezkoa da erabiltzaileek chatbot-a probatzeko garaian protokolo bat jarraitzea.

Lehenik eta behin, albisteak lortzeko moduluak ez du arazorik eman, eta Berria egunkariaren webguneak HTML kode eta egitura finkoa duenez, modulu horrek emaitza onak ematen jarraituko duela ondoriozta daiteke. Hori dela eta, modulu hori ez da kontuan hartu protokoloa diseinatzean.

Protokoloaren lehenengo pausua, erabiltzaileek dialogo-sistema probatu aurretik, egun horretako Berria-ko webgunea aztertzea izan da, egun horretan zein berri dauden jakin ahal izateko. Horrela, webgunean azaltzen diren bi albiste eta bi kategoria ausaz aukeratu dira. Horrez gain, sistemaren indizeak egun horretako berriekin eguneratu dira, webguneak eta indizeak eduki berdinak izan ditzaten. Ebaluazioa bost egun desberdinetan egin da, bilatzeko eskatzen diren albisteak eta kategoriak desberdinak izan daitezten.

Ondoren, ausaz aukeratutako bi albiste eta bi kategoriei buruzko informazioa lortzeko eskatu zaie erabiltzaileei. Horretaz gain, gai libreko albiste bat bilatzea ere eskatu zaie. Hori amaitzean, erabiltzaileek chatbot-arekin egindako prozesu berdina Berria-ko webgunean errepikatu dute.

Hori guztia egindakoan, ondorengo galderak dituen galdetegi bat bete dute erabiltzaileek:

1. Bost bilaketa desberdin egin dituzu, zenbatetan lortu dituzu albiste esanguratsuak?
2. Itzulitako albisteak alde batera utzita, izan ditudan elkarrizketa guztietan chatbot-ak zentzuzko gauzak esan ditu.
3. Aurkitu nahi nituen albisteak Berriako web orrian baino azkarrago aurkitu ditut.
4. Chatbot hau erabiliko nuke egunerokotasunean albisteak bilatzeko.
5. Arazoren bat izan baduzu, edo hobekuntzaren bat baldin baduzu, hemen adierazi.

Galderei dagokienez, lehenengo galderan 0-tik 5-erako zenbaki bat aukeratu behar da, bigarren, hirugarren, eta laugarren galderetan 1-etik 5-erako zenbaki bat aukeratu behar da, non 1-ek guztiz desados eta 5-ek guztiz ados errepresentatzen duten, eta azken galdera testu bidez erantzuteko galdera bat da. Sortutako ebaluazio-galdetegia [B.1](#) Irudian ikus daiteke.

Hontaz guztietaz gain, erabiltzaileek izan dituzten elkarrizketa guztiak erregistratuta gelditu dira (elkarrizketa bakoitza fitxategi desberdin batean). Horrela, ebaluagarriak diren bi elementu nagusi lortu dira ebaluazio prozesua egin ostean: galdetegiaren erantzunak eta elkarrizketak.

Galdetegiko erantzunetatik, chatbot-aren funtzionamenduari buruz esanguratsua den informazio eta ondorioak lor daitezke. Hala ere, erabiltzaileek emandako erantzunak subjektiboak direnez, ezin dira erantzun horietatik balio kuantitiboak lortu. Hori dela eta, erantzun horiek informazio kualitatiboa lortzeko erabili dira.

Bestalde, erabiltzaileen elkarrizketetan garrantzitsuak diren bi faktore ikus daitezke. Bate-tik, erabiltzaileen mezuak intent egokiarekin sailkatu eta horren baitan, behar den akzioa exekutatu dela beha daiteke, eta bestetik, erabiltzaileak egindako informazio eskaerareki-ko sistemak itzulitako albisteak esanguratsuak diren ala ez ikus daiteke.

Horrela, elkarrizketa guztietatik ausaz berrogei mezu hartu dira eta mezu horien intent-a ondo sailkatu den eta dialogo-sistemak exekutatu duen akzio zuzena den ala ez ikusi da. Bestalde, egindako albisteen eskaera guztietatik ausaz hamar eskaera desberdin hartu dira, eta erabiltzaileak albiste esanguratsuren bat jaso duen ala ez ikusi da.

Mezu baten intent-a zuzena izateko, intent honek mezuaren intentzioarekin bat etorri behar du. Adibidez, sarrerako mezua “Laguntza behar dut” baldin bada, eta mezuari eskaera intent-a esleitzen bazaio, esleipena okerra izan dela kontsideratuko da, mezuaren benetako intentzioa laguntza eskatzea zelako.

Bestetik, erabiltzaileak jasotako mezuaren intent-arekiko zentzuduna baldin bada, chatbot-ak iragarritako akzioa zuzena izango da. Esate baterako, “Laguntza behar dut” mezua ja-sota, chatbot-ak agurtzeko mezua bidaltzen badio, akzioa gaizki iragarri dela esango da, akzio zuzena laguntza mezua bidaltzea zelako. Hontaz gain, intent-a gaizki sailkatu den kasuetan, chatbot-ak exekutatu duen akzioa intent okerrarekiko zentzuduna bada, akzioa zuzena dela kontsideratu da.

Bilaketa-sistemari dagokionez, emaitzarik ez bada itzuli edota itzulitako emaitzek egin den eskaerarekiko zerikusirik ez badute, sistemak egindako bilaketa okerra kontsideratu da. Modu berean, eskaera baten ondoren itzulitako emaitzen artean, gutxienez interesga-rria den albiste bat baldin badago, bilaketa hori zuzentzat hartu da.

Amaitzeko, chatbot-a ebaluatzeko bi elementu horiek konbinatzeak, sortutako dialogo-sistemari eta bilaketa-sistemari buruzko ikuspegi zabalago bat emango du, proba garaian azaldu diren arazoak aztertzeko, eta bertatik hobekuntza eta etorkizunerako lan posibleak proposatu ahal izateko.

4.2 Emaitzak

Dialogo-sistema ebaluatzeko sortu den protokoloarekin, ebaluazioa lagin handi batekin egin zitekeen arren, elkarrizketak eskuz aztertu behar izateak, denbora kopurua eta elka-rrizketak ebaluatuko dituzten pertsona kopurua handiak izatea eragiten du. Kasu honetan baliabide falta dagoenez, chatbot-a zortzi pertsonen probatzea erabaki da.

Ekintza	Kopurua
Intent sailkapen zuzenak	32/40
Iragarritako akzio zuzenak	40/40
Bilaketa zuzenak	6/10

4.1 Taula: Ebaluazioaren emaitza kuantitatiboak.

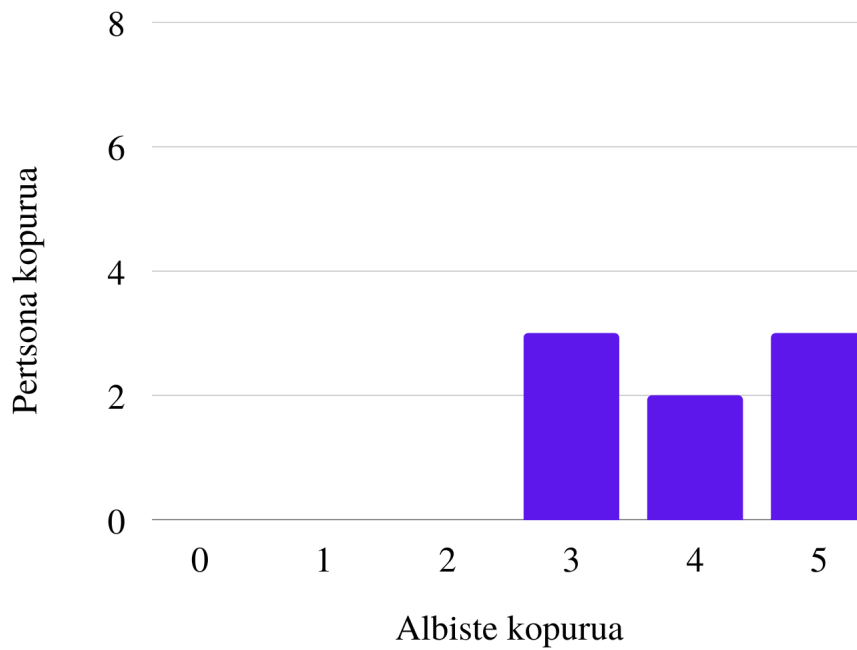
Erregistratuta gelditu diren zortzi elkarrizketa horietatik, elkarrizketa bakoitzetik erabiltzaileak idatzitako bost mezu, mezu hauei esleitutako intent-ak eta mezu horiei chatbot-ak emandako bost erantzunak hartu dira (elkarrizketa bakoitzetik bost mezu hartu direnez, guztira berrogei mezu eskuz ebaluatu dira, protokoloan definitu den bezala).

Ondoren, erabiltzaileen mezuak eta horien intent-ak hartuta, Rasa-ren modeloak mezu bakoitzari esleitutako intent-a zuzena den ala ez aztertu da. Gainera, emandako bost erantzunekin, modeloak akzio edo erantzun zuzena iragarri duen ikusi da. Bestetik, bilaketa-sistemaren kalitatea neurtzeko, egin diren bilaketa guztietatik ausaz hamar bilaketa hartu dira, eta sistemak itzuli duen emaitza aztertu da.

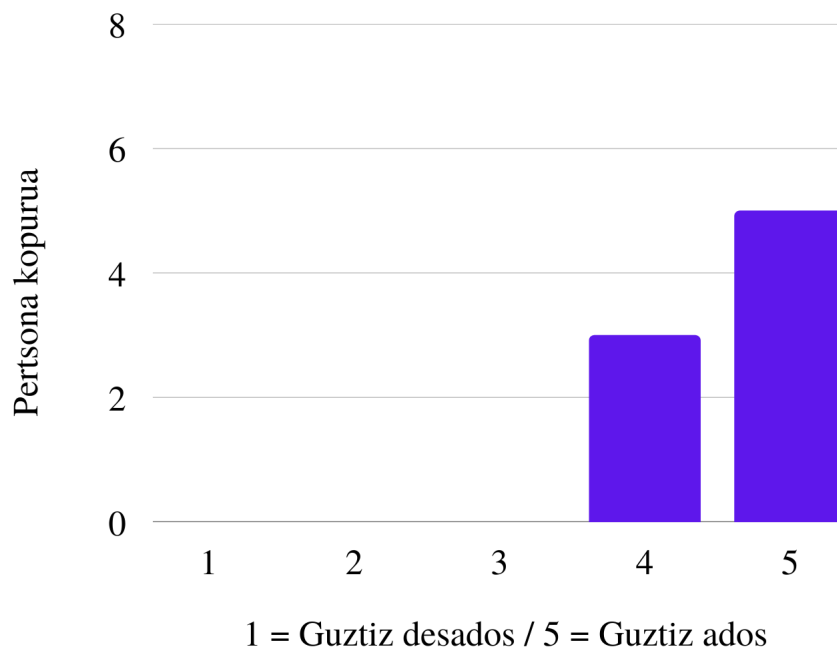
Hori guztia kontutan hartuta, 4.1 Taulak ebaluazioaren emaitza kuantitiboak jasotzen ditu. Bertan iragarritako intent eta akzio zuzenen kopurua, eta bilaketa-sistemak egindako bilaketa zuzenen kopurua azaltzen dira. Bestalde, galdetegian lortutako emaitzak grafiko-ki errepresentatu dira, horiek errazago interpretatzeko.

Orokorrean, lortutako emaitzak onak direla esan daiteke, intent sailkapenak kasu partikular batzuetan izan ezik, ondo funtzionatzen du, eta exekutatu beharreko akzio guztiak ondo iragarri dira. Bilaketa-sistemari dagokionez, beranduago aztertuko diren zenbait arazo aurkeztu ditu proba garaian.

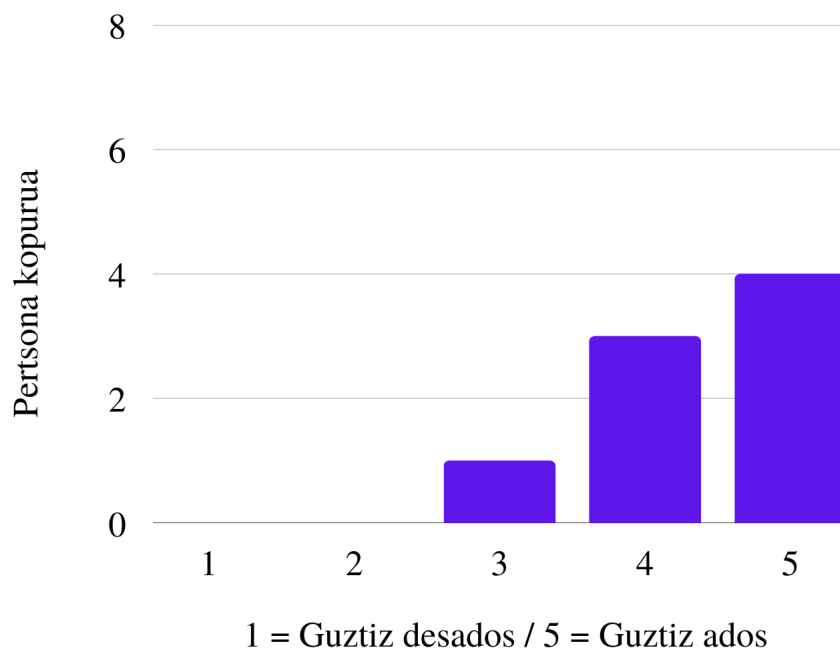
Azkenik, erabiltzaileek chatbot-arekin izan duten interakzioa ulerterraza eta sinplea izan dela esan daiteke 4.2 Irudia begiratuta, eta argibide gutxi batzuekin chatbot-arekin arazorik gabe hizketan hasteko gai izan dira. 4.3 Irudiko grafikoa erreparatuz, orokorrean sortutako dialogo-sistema erabiltzeko eroso dela eta albisteak honen bidez bilatzea Berriko webgunera joatea baino azkarragoa dela ondorioztatu daiteke. Gainera, erabiltzaileek egindako eskaera gehienetan interesgarria den berriren bat jaso dute (4.1 Irudia). Hala ere, 4.4 Irudiaren arabera ez dago garbi ea erabiltzaileek chatbot hau egunerokotasuean erabiliko zuten ala ez (beste modu batera ulertuta, ea chatbot hau Berria egunkariaren webgunearen ordeztu erabiliko zuten).



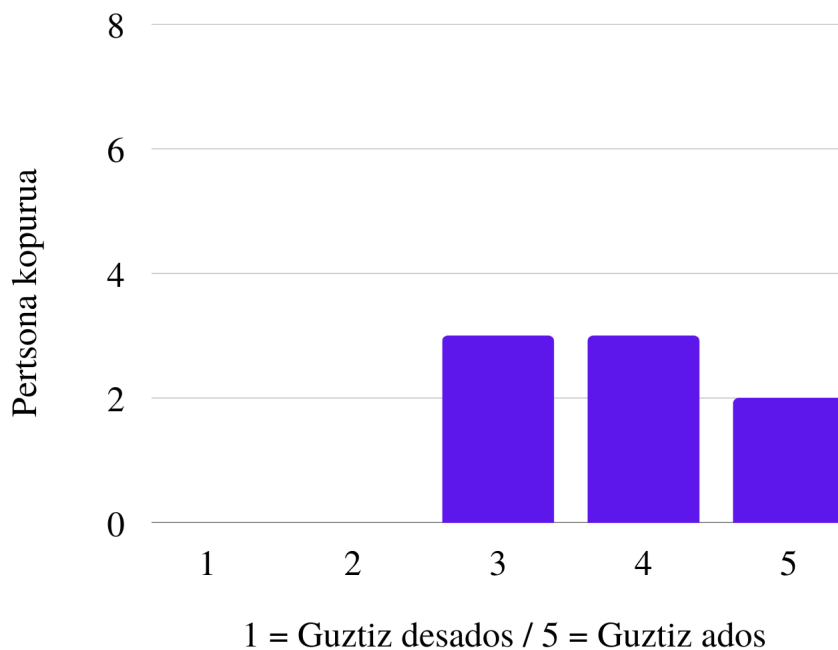
4.1 Irudia: “5 bilaketa desberdin egin dituzu, zenbatetan lortu dituzu albiste esanguratsuak?” galderaren emaitzak.



4.2 Irudia: “Itzulitako albisteak alde batera utzita, izan ditudan elkarrizketa guztietan chatbot-ak zentzuzko gauzak esan ditu” galderaren emaitzak.



4.3 Irudia: “Aurkitu nahi nituen albisteak Berriako web orrian baino azkarrago aurkitu ditut” galderaren emaitzak.



4.4 Irudia: “Chatbot hau erabiliko nuke egunerokotasunean albisteak bilatzeko” galderaren emaitzak.

USER: Boris Johnson

CHATBOT: Kaixo! Ni Juliet chatbot-a naiz.

CHATBOT: Berria egunkariko artikuluak modu errazago batean bilatzeko programatu naute. Bilaketak egiteko, "hitz_gakoak albisteak" moduko formatua erabili, mesedez. Adibidez: "Futboleko albisteak". Zalantzak badituzu, "laguntza" idatzi.

USER: Boris Johnson-i buruzko albisteak

CHATBOT: 1) Boris Johnsonek zentsura mozioari aurre egin beharko dio gaur bere alderdian

4.5 Irudia: Intent sailkapen okerra duen elkarrizketa.

4.3 Eztabaida

Chatbot-aren funtzionamendua hobeto ulertzeko, beharrezkoa da lortutako emaitza guztiak modu sakonago batean aztertzea.

Intent detekzioa

Intent sailkapenaren aldetik, arestian esan bezala, kasu konkretu batzuetan akatsak egiten ditu. Izan ere, albiste baten eskaera egitean, erabiltzaileei antzeko formatu bat mantentzeko eskatzen zaie (adibidez, "Futboleko berriak") ondoren modeloarentzat esaldi hori informazio eskaera bat dela identifikatzea errazagoa izan dadin. Nahiz eta mezuak idazteko formatu hori erabiltzaileentzat erosoena ez izan (erosoagoa izango litzateke bilatu nahi den albistearen hitz gakoak bakarrik idaztea), intent detekzioa asko errazten du eta emaitzak hobetzen ditu. Are gehiago, erabiltzaileek albisteak bilatzeko soilik hitz gakoak erabili dituztenean, mezu horien intent-a gehienetan gaizki sailkatu da.

4.5 Irudian ikus daitekeen elkarrizketan, erabiltzailea Boris Johnson-i buruz bi modu desberdinetan berriak bilatzen saiatu da, baina bakarrean lortu du nahi zuen emaitza. Izan ere, lehenengo saiakerako mezuan, soilik bilaketaren hitz gakoak erabili ditu, eta modeloak mezu horren intent-a agurtzea dela interpretatu du. Ondorioz, modeloak erabiltzailea bueltan agurtu du. Bigarren saiakeran, berriz, erabiltzaileak eskatutako formatu erabili du, eta modeloak mezu hori informazio eskaera bat bezala sailkatu du. Hori dela eta, chatbot-ak erabiltzailearen mezutik hitz gakoak atera, bilaketa egin eta lortutako artikulua itzuli dio.

Bestalde, Rasa-k entrenamendu datu kopuru txikiarekin chatbot bat sortzeko aukera ematen du. Garatzaileentzat interesgarria da hori, askotan zaila delako errealak diren entre-

namendu datuak sortzea, baina desabantaila ere bihur daiteke, dialogo-sistemak entrenamendu garaian ikusi ez duen esaldi bat sarrera moduan jasotzeko probabilitatea handitu egiten delako. Hau gertatuz gero, modeloak ez du jakingo esaldi horren intent-a zein den, eta ondorioz, emango duen erantzuna ez da zuzena izango.

Halaber, kontuan izan behar da chatbot-aren modelo sortzeko pipeline bat sortzen dela. Pipeline-ko elementuak ez daude euskararako pentsatuta, eta litekeena da chatbot berdina ingeleserako edo gaztelararako sortuz gero, intent detekzioa hobea izatea.

Iragarritako akzioak

Dialogo-sistemak mezu baten intent-a lortzen duenean, intent horren arabera zein akzio exekutatu behar duen (hau da, erabiltzaileari zein erantzun emango dion) erabaki behar du. Kasu honetan, emandako erantzun guztiak zuzenak izan dira. Modeloak mezu baten intent-a gaizki sailkatu duen kasuetan, exekutatu beharreko akzioa intent oker horren arabera izan da, ondorioz, akatsa intent detekzioan egon da, ez iragarritako akzioan.

Bilaketa-sistema

Bilaketa-sistemari dagokionez, orokorrean emaitza onak eman ditu, baina arazo batzuk ere identifikatu dira.

Lehenik eta behin, bilaketa batekiko emaitzarik ez jasotzeak bi arrazoiren ondorioz gertatzen dela ikusi da. Alde batetik, erabiltzailea bilatzen saiatzen ari den albistea ez da indizean existitzen. Chatbot-ak itzultzen dituen berri guztiak, Berria egunkaritik hartzen dira, hori dela eta, erabiltzaileak nahi duen albistea egunkarian ez badago, ezinezkoa izango da hau aurkitzea, albistea ez baita indizean egongo. Bestetik, erabiltzailea bilatzen saiatzen ari den albistea indizean existitzen da, baina bilaketa sistemak ez du aurkitu.

Bilaketetan eragin zuzena duen lehenengo faktorea, bilaketaren hitz gakoak idazteko modua da. Edozein akats ortografiko, idazteko modu edo hitzen aldaerek, bilaketan eragina izango dute, eta litekeena da bilatzen ari diren albisteak ez jasotzea. 4.6 Irudiko elkarrizketa horren adibidea da. Elkarrizketa horretan argi ikusi daiteke, “Errusia” hitza gaizki idazteak, existitzen ziren albisteak ez lortzea ekarri duela.

Bestetik, kategoria gramatikalaren etiketatzeak ere eragina du bilaketak egiterakoan. Demagun izen arrunta den hitz bat aditzondo moduan etiketatzen dela. Bilaketak egiterakoan, izenak eta adjektiboak ez diren hitz guztiak bilaketaren katetik ezabatzen direnez,

USER: Rusia albisteak

CHATBOT: Ez dut artikulurik topatu arlo horren inguruan.

USER: Errusia albisteak

CHATBOT: 1) Stoltenbergen esanetan, Putinek «hutsegite handi bat» egin du
2) Errusiako petrolioaren enbargoa blokeatzen jarraitzen du Hungariak
3) Ekialderako bidean

4.6 Irudia: Bilaketa sistemaren emaitza okerra duen elkarrizketa.

bilaketarentzat esanguratsua zen hitz horrek ez du eraginik izango bilaketean. Ondorioz, lortutako emaitza zuzena ez izatea gerta liteke.

Honetaz guztietaz gain, bilaketa-sistema honek ez du bilaketaren testuingurua kontutan hartzen bilatzeko garaian. Hau da, demagun gure indizean Errealaren azken futbol partidari buruzko albisteak daudela, baina albiste horietan ez dela “Futbol” hitza azaltzen. Erabiltzaileak “Futboleko albisteak” eskaera egiten badu, ez du albisterik jasoko, BM25 algoritmoak *exact matching* teknika erabiltzen duelako, hau da, eskaeraren hitzak dokumentuan dauden ala ez bilatzen du, eskaera eta dokumentuen testuinguruari inolako garrantziarik eman gabe.

Azkenik, bilaketa berean bi gairi buruzko albisteak bilatzea posible da. 4.7. Irudiko elkarrizketan, erabiltzaileak bi gai jarri ditu eskaeran, eta bi gai horiei buruzko artikulak lortu ditu. Funtzionaltasun hori ez zegoen planteatuta chatbot-aren hasierako diseinuan (bilaketak banan-banan egiteko pentsatuta dago), baina egindako proben bidez hau egitea posible dela ikusi da.

Erabiltzaileen emaitzak

Erabiltzaileei degokienez, chatbot-a erabilgarria eta erabiltzeko eroso dela esan daiteke. Horrez gain, argibide gutxi batzurekin erabiltzaile guztiak dialogo-sistema arazorik gabe erabiltzeko gai izan dira, eta inork ez du arazorik izan bilaketak egiteko edo albisten baten edukia lortzeko.

Chatbot-aren zerbitzaria ikaslearen ordenagailuan exekutatu da, eta hala ere erabiltzaileek izandako erantzun denborak oso onak dira. Hori dela eta, chatbot hau baliabide gehiago dituen makina batean erabiltzaileentzat publikoki jarriko balitz, egindako eskaerak arazorik gabe zerbitzatzeko gai izango litzatekeela ondoriozta daiteke. Kontuan izan behar da, zerbitzariak emandako erantzun-denborak luzeegiak badirela, chatbot-a ez dela erabilgarria izango, nahiz eta emandako emaitza guztiak zuzenak izan.

USER: Liverpool eta Kolonbiako albisteak

CHATBOT: 1) Mendekua batek, balentria besteak

2) Petro eta Hernandez lehiatuko dira Kolonbiako bozen bigarren itzulian

3) EZKERRA, AGINTEA ESKURATZEKOTAN

USER: Lehenengoa

CHATBOT: Liverpoolen eta Real Madrilen Txapeldunen Ligako finala jokatuko

dute gaur, Parisen. Hirugarren aldiz ariko dira elkarren aurka final batean:

1981ean, Merseysideko taldeak irabazi zuen, eta 2018an, berriz, Madrilgoak.

[https://www.berria.eus/paperekoa/1872/021/001/2022-05-28/mendekua-](https://www.berria.eus/paperekoa/1872/021/001/2022-05-28/mendekua-batek-balentria-besteak.htm)

[batek-balentria-besteak.htm](https://www.berria.eus/paperekoa/1872/021/001/2022-05-28/mendekua-batek-balentria-besteak.htm)

USER: 2

CHATBOT: Aurreikusitako kopuruetatik gertu ibili da Petro: botoen %40 lortu ditu.

Sorpresarik handiena Rodolfo Hernandez hautagai populistaren eskutik etorri

da: botoen %28 lortu ditu. Federico Gutierrez eskuineko hautagaiak egurra

jaso du: botoen ia %24 lortu ditu.

[https://www.berria.eus/albisteak/214042/petro-eta-hernandez-lehiatuko-](https://www.berria.eus/albisteak/214042/petro-eta-hernandez-lehiatuko-dira-kolonbiako-bozen-bigarren-itzulian.htm)

[dira-kolonbiako-bozen-bigarren-itzulian.htm](https://www.berria.eus/albisteak/214042/petro-eta-hernandez-lehiatuko-dira-kolonbiako-bozen-bigarren-itzulian.htm)

4.7 Irudia: Bi gaiei buruzko eskaera duen elkarrizketa.

Bestalde, chatbot-aren funtzionamendua orokorrean egokia izan dela ondoriozta daiteke 4.1, 4.2 eta 4.3 Irudiei erreparatuz. Albisteak bilatzerako garaian, gehienetan interesgariak diren artikulua itzuli ditu, emandako erantzunak orokorrean zentzudunak izan dira eta albisteak bilatzeko prozesua azkarra eta sinplea da. Hori ikusita, lan honetako helburu nagusiak lortu direla esan daiteke: berriak lortzeko tresna berri eta sinple bat sortzea, chatbot-ak erabiliz.

Azkenik, 4.4. Irudiko emaitzetan oinarrituta, ez dago guztiz argi ea erabiltzaileek chatbot hau Berria egunkariaren webgunearen ordean erabiliko zuten ala ez. Berriaren webgunean albiste askoren titularrak begirada batez ikus daitezke, hori dela eta, erabiltzaileak segundu gutxi batzuetan egun horretan gertatu diren notizien ideia bat izan dezake, hau da, egunaren laburpen bat jaso dezake. Ondorioz, zehazki zer bilatu ez dakien erabiltzaile batentzat, webgunea chatbot-a baino erosoagoa suertatu daiteke, azken horrek ez baititu albiste posible guztiak bistaratzen. Dialogo-sistema, berriz, gai konkretu bati buruz informazioa azkar jakiteko aproposagoa izan daiteke erabiltzaileentzat. Horrez gain, Telegram baino ospe handiagoa duen beste kanal bat erabiltzeak, erabiltzaileentzat chatbot-a erabiltzea errazagoa izatea ekar dezake.

5. KAPITULUA

Ondorioak eta etorkizuneko lana

Lan honetan, Berria egunkariko albisteak emateko gai den Juliet chatbot-a garatu da eta Telegram aplikazioarekin integratu da. Horretarako, lehenik eta behin beharrezkoak diren kontzeptu teorikoak eta implementaziorako erabili diren tresnak landu dira. Ondoren, chatbot-a garatu da, horren implementazioa bi zati desberdinetan banatuz: Rasa-ren bidez egindako dialogo-sistema eta albisteen bilaketa-sistema. Azkenik, chatbot-aren errendimendua ebaluatu da. Hori egiteko, lehenik ebaluazio-protokolo bat definitu da, ondoren ebaluazio-protokolo hori zortzi pertsonetako lagin batean jarraitu da eta azkenik lortutako emaitzak aztertu dira.

Atal honen helburua proiektuaren ondorioak, lan hau egitetik izandako ondorio pertsonalak eta etorkizunerako lanak azaltzea da.

5.1 Proiektuaren ondorioak

Lehenik eta behin, aipatu beharra dago ebaluzio-protokolotik ateratako emaitzak oso positiboak izan direla: chatbot-aren funtzionamendua esperotakoa izan da kasu gehienetan, erabiltzaileak behar zituzten albisteak aurkitzeko gai izan dira eta, orokorrean, albiste konkretu bat bilatzeko chatbot-a erabiltzea webgunera joatea baino azkarragoa dela ondorioztatu da. Hala ere, ebaluazio-laginaren erantzunetatik ez da argi gelditzen eguneroko informazio-iturri bezala erabiliko luketen ala ez.

Rasa-ri dagokionez, chatbot-ak sortzeko oso tresna ahaltsua dela esan behar da. Izan ere,

tresna horrek erabilera pertsonalerako edo produktu komertzial bat sortzeko balio dezake. Inplementaziorako eskaintzen dituen elementu desberdinak, dokumentazio zabalak eta Rasa-k berak duen komunitateak, lan-ingurune horrekin lan egiten hastea zein konplexutasun handiko proiektuak garatzea ahalbidetzen du.

Bestetik, azpimarratu behar da garatutako bilaketa-sistemak emaitza onak eman dituela eskuragarri izandako baliabideak kontsideratuz. Nahiz eta bere mugak izan, inplementatzeko eta integratzeko erraztasuna eta bilaketetan eman dituen emaitzak ikusita, proiektu honen tamainarentzat oso egokia den sistema bat sortu dela esan daiteke. Izan ere, ohiko kasuetan beti interesgarriak diren albisteak itzuli ditu.

Erabiltzaileek Chatbot-arekin izandako interakzioaren asebetetze-maila ere aipagarria da. Chatbot bat garatzean, kontuan izan behar da erabiltzaileek chatbot-arekin izango duten elkarrekintza, eta inplementazio guztiak hori izan behar du oinarri bezala. Chatbot-a erabiltzea zaila edo intuitiboa ez bada, berdin da atzetik zein bilaketa-sistema jartzen den, erabiltzaileek ez baitute jakingo hau erabiltzen. Hori dela eta, garrantzitsua da xehetasun guztiei garrantzia ematea, ondoren sortu nahi den produktua erabilgarria izatea nahi baldin bada. Kasu honetan, ebaluzio-protokoloak argi utzi du garatutako chatbot-a aurretik ezagutzarik gabeko erabiltzaileentzat ere eroso eta aproposa dela.

Amaitzeko, ondorio nagusi bezala, proiektu honen benetako indarra sortu diren hiru moduletan dagoela esan behar da. Izan ere, nahiz eta hiru moduluak haien artean komunikatu, modu independentean garatu dira. Horrela, inplementazio guztia aldatu gabe, chatbot-ak ematen duen informazioaren domeinua, erabiltzaileekin hitz egiteko modua edo bilaketak egiteko prozesua hobe daiteke.

Alde batetik, *scraping* egiteko modulua aldatuz, chatbot-ak eskaintzen duen informazioa alda daiteke. Esate baterako, soilik albisteen azpтитulua eta esteka eman beharrean, zenbait aldaketa eginez chatbot-ak datu gehiago eman ditzake. Adibidez, albiste osoa ematea nahi bada, hau webgunetik hartu eta indizean gordetzearekin nahikoa izango litzateke. Are gehiago, Juliet-en domeinua guztiz alda zitekeen aipaturiko modulua aldatu eta gainontzeko moduletan moldaketa minimoak eginez. Egunkari bateko albisteak eman beharrean, eguraldiaren inguruko informazioa jasotzeko, Wikipedia-ko¹ artikuluak irakurtzeko edota bestelako multimedia-edukia biltzeko gai den *scraper* bat inplementatuz gero, beste esparru batzuetan erabilgarriak izan daitezkeen chatbot-ak erraz inplementa daitezke lan honetan oinarrituta.

Bilaketa-modulua ere nahiko malgua izan daiteke erabiltzaileen eskaerak tratatzerako ga-

¹<https://eu.wikipedia.org/wiki/Azala>

raian. Esate baterako, BM25 algoritmoaren aldaera desberdinak implementa daitezke edo informazio-erazketarako erabiltzen diren beste arkitektura batzuekin integra daitezke. Aldaketa horiek lan hau bere baitan hobetzeko helburua izan dezakete edo, lehen aipatu bezala, beste izaera bateko datuekin lan egitea ahalbidetu dezakete.

Chatbot-ari berari dagokionez, erabiltzaileekin dituen elkarrekintzak ere alda daitezke, haien behar eta egoera ezberdinetara egokituz. Adibidez, Juliet-ekin idatziz hitz egin behar denez, nahiko deserosoa izan daiteke pertsona adinduentzat. Modu horretan, chatbot honekin ahotsaren bidez komunikatzeko aukera emanez gero, zenbait erabiltzaileentzat erosoagoa izango litzateke. Horretaz gain, chatbot-a garatzeko Rasa erabili den bezala, beste *framework* batzuk ere erabil daitezke, hala nola, Wit.ai² edo Botpress³.

Laburbilduz, sortutako dialogo-sistemak hasieran planteatutako helburuak guztiz bete dituzte. Gainera, garatutako arkitektura oso malgu da eta proiektu berriak sortzeko eredu posible bat erakusten du.

5.2 Ondorio pertsonalak

Pertsonalki, proiektu hau egitea esperientzia aberasgarria izan da eta garapenaren zehar ikasitako guztia nire etorkizunerako oso lagungarria izango dela deritzot. Izan ere, inplementazioaz gain, ezagutzen ez nituen kontzeptuetan sakontzen, desbiderapenak kudeatzen, dokumentazio teknikoa idazten eta proiektuaren kontrol bat eramaten ikasi behar izan dut.

Proiektu hau hasi aurretik, ez nuen inoiz hizkuntzaren prozesamenduko arloan lan egin. Gainera, nahiz eta proiektu hau berez chatbot bat izan, hizkuntzaren prozesamenduko arlo desberdinen beharra azaldu da, inplementazioa aurrera joan ahala. Adibidez, testua prozesatzeko teknika desberdinak edo bilaketa sistemak sortzeko oinarriak. Horrela, proiektu hau amaitzean, hizkuntzaren prozesamenduko ikuspegi oso zabala lortu dudala esan dezaket. Hontaz gain, proiektu hau egin ahal izateko, informatikaren arlo desberdinetako kontzeptuak behar izan dira, adibidez, sareetako eta software ingeneritzako kontzeptuak. Oso interesgarria iruditu zait atal desberdinak gutxinaka inplementazioan azaltzea.

Beste alde batetik, proiektu bat garatzean eskuragarri dauden baliabideak, makinak eta pertsonen aldetik, ondo neurtzearen garrantzia ikasi dut. Funtzionaltasun berri bat gehitu nahi izateak, proiektuaren beste esparru batzuei energia gutxiago ematea ekar dezake, eta

²<https://wit.ai/>

³<https://botpress.com/>

ondorioz, funtsezkoak izan daitezken atalek kalitate baxuagoa izan dezakete. Hori dela eta, hobe da helburu gutxi batzuetan indar guztiak jartzea, eta hauek kalitate handiarekin egiten direla ziurtatzea, kalitate ertaineko helburu asko aldi berean egiten saiatzea baino.

Amaitzeko, ondorio orokor moduan, proiektu hau gradu guztian zehar ikusitako kontzeptuak erabiltzeko, arlo berri batean lan egiteko eta pertsonalki hazteko aukera paregabea izan dela deritzot.

5.3 Etorkizunerako lana

Proiektua garatu eta ebaluatu ostean, honek erakutsi dituen gabeziak hobetzeko asmoz, zenbait hobekuntza edo proiektu osagarri egin daitezke, egindako dialogo-sistemaren kalitatea hobetzeko. Hona hemen proposatutako etorkizunerako lan desberdinak:

- Chatbot-a beste domeinu batzuetara egokitzea:

Ondorioetan aipatu den bezala, *scraping* egiteko modulua aldatuz, Juliet beste domeinu batzuetara egoki daiteke arazo handirik gabe. Hori dela eta, eguraldia edo Wikipedia-ko artikuluak lortzeko modulu hori egokitzea interesgarria izan daiteke.

- Chatbot-ari funtzionaltasun berriak gehitzea:

Sortu den chatbot-ak honen diseinuan definitutako bi funtzionaltasunak arazorik gabe betetzen ditu. Hala ere, chatbot orokorrago bat sortzeko, interesgarria izango litzateke egunkari desberdinetatik berriak hartzea edo hemeroteca moduko bat gordetzea. Egun desberdinetako albisteak emateko gai izanez gero, egunaren bidez bilaketak egitea ahalbidetu beharko litzateke.

- Rasa-X martxan jartzea:

Rasa-X chatbot-ak ebaluatzeko Rasa-k eskaintzen duen tresna bat da. Tresna honen bitartez, erabiltzaileek dialogo-sistemarekin dituzten elkarrizketak ikusi, intent sailkapenaren ebaluazioa egin eta chatbot-a erabiltzaileen artean partekatzea errazago egin daiteke. Funtsean, ebaluazio ataza asko errazten du. Tresna hau martxan jartzen saiatu naiz, baina horretarako sareetako eta *cluster*-en ezagutza behar da. Rasa-X egindako dialogo-sistemarekin konektatzeak, ebaluazio modu zehatzago batean ekarriko luke.

- Rasa beste hizkuntza batekin probatzea:

Ebaluazio atalean intent detekzioa chatbot hontan hobetzeko gelditu den ataza bat dela ikusi da. Rasa-ren modeloa entrenatzeko elementuak ez daudenez euskararako pentsatuta, interesgarria izan daiteke gaztelaniazko ala ingelesezko antzeko dialogo-sistema bat sortzea, ondoren intent detekzioaren eta NLU moduluaren benetako ahalmena ikusi ahal izateko.

- Bilaketa sistema hobetzea:

Sortutako dialogo-sistemak, ez du eskaeraren testuingurua kontutan hartzen eta eskaera idatzi den moduaren arabera lortutako emaitzak desberdinak izan daitezke. Hau konpontzeko, interesgarria izan daiteke sortutako bilaketa sistema DPR (*Dense Passage Retrieval*) [Karpukhin et al., 2020] sistema batekin trukitzea. Honen bidez, bilaketetan izandako emaitzak hobetu beharko lirateke. Hala ere, DPR bat martxan jartzeko, beharrezkoa da hardware aldetik baliabide asko izatea.

- Ebaluazio protokolo berri bat definitzea:

Ebaluaziorako erabili den protokoloak, gordetako elkarrizketak eskuz errepatatzea ekartzen du. Honen ondorioz, baliabideen kostua igo egiten da eta gainera giza erroreak egoteko arriskua sortzen du. Horregatik, interesgarria izango litzateke chatbotak ebaluatzeko metodo automatiko bat sortzea, ebaluazioa errazteko asmoz.

Eranskinak

A. ERANSKINA

Proiektuaren helburuen dokumentua

A.1 Proiektuaren deskribapena eta helburuak

Proiektu honen helburua, eguneko albisteak emateko gai den chatbot bat garatzea eta hau Telegram aplikazioarekin integratzea da, Rasa *framework*-a erabiliz. Hau egin ahal izateko, beharrezkoa da erabiliko diren tresna eta liburutegiak erabiltzen ikastea.

A.2 Proiektuaren plangintza

A.2.1 LDE diagrama

Proiektuaren garapenaren zehar, ataza desberdinak banatu eta deskonponatu egin dira Lanaren Deskonposaketa Egitura (LDE) diagrama bidez ([A.1](#) Irudia).

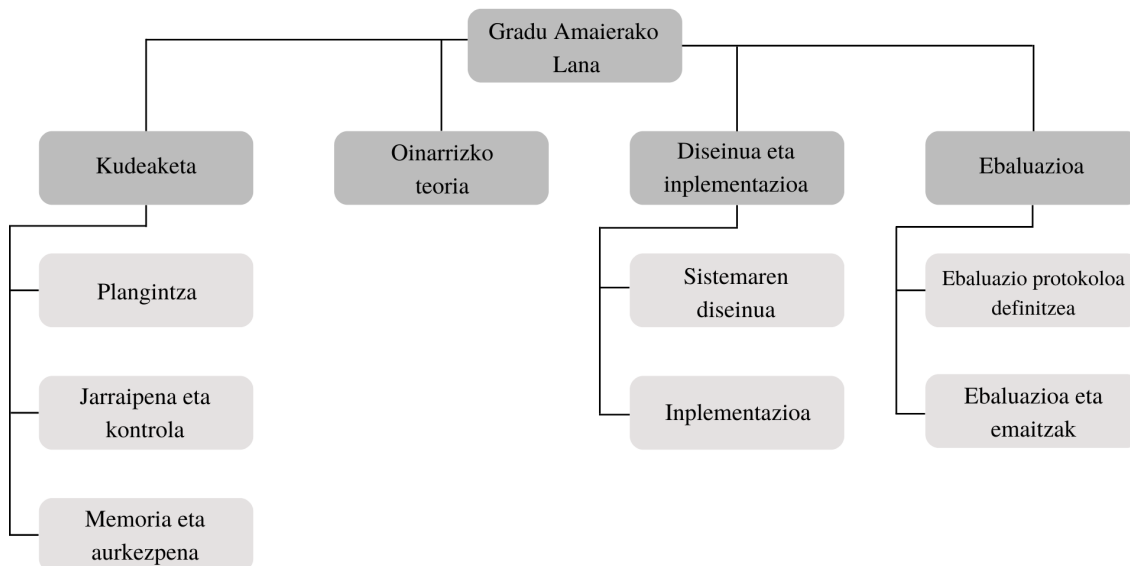
A.2.2 Lan - paketeak

Plangintza

Atal honetan, proiektu honen plangintza egin da. Horretarako, proiektuaren helburu nagusiak, ataza desberdinak, mugarriak, proiektuaren amaieran sortuko diren emangarriak eta

Lan-paketea	Iraupena (orduak)
Kudeaketa	135
Plangintza	10
Jarraipena eta kontrola	35
Memoria	90
Oinarrizko teoria	90
Diseinua eta implementazioa	160
Sistemaren diseinua	40
Implementazioa	120
Ebaluazioa	28
Ebaluazio protokoloa definitzea	3
Ebaluazioa eta emaitzak	25
GUZTIRA	413

A.1 Taula: Lan-pakete bakoitzari esleitutako ordu kopurua



A.1 Irudia: LDE diagrama

lan metodologia nolakoa izango den definitu da. Hortaz gain, proiektuaren bideragarritasuna, arriskuak eta prebentzioa ere aztertu dira.

Jarraipena eta kontrola

Jarraipena eta kontrola proiektuaren garapen osoaren zehar egin beharreko ataza bat izango da. Izan ere, garrantzitsua da proiektuaren hasieran jarritako helburu eta mugarriak betetzen direla ziurtatzea. Hau bermatzeko, izandako desbiderapenak uneoro jasoko dira eta hauei aurre egiteko plangintza aldatzea gertatu liteke. Hortaz gain, kontrol hau eraman ahal izateko jarraipen bilerak egingo dira periodikoki.

Memoria eta aurkezpena

Atal honetan, proiektu honen eduki guztiak gordeko dituzten memoria eta aurkezpena egingo dira.

- Memoria: Egindako proiektuaren informazio guztia duen dokumentua.
- Aurkezpena: Proiektuaren defentsa egiteko, lagungarria da aurkezpen bat erabiltzea. Hontan, egindako lanari buruzko puntu garrantzitsuenak azalduko dira, modu argian eta antolatuan.

Oinarrizko teoria

Atal honen helburua, proiektua garatzeko erabili diren tresnen atzean dagoen oinarri teorikoa sakonki ulertzea da. Hauen artean, *web scraping*, tokenizazioa, lematizazioa, kategoria gramatikalen etiketatzea, informazio erauzketa eta bm25 algoritmoa eta dialogo-sistemak aztertu dira.

Sistemaren diseinua

Sortuko den dialogo-sistemaren eta berriak bilatzeko sistemaren xehetasun guztiak definituko dira, ondoren garatuko den implementazioa egindako diseinuan oinarritu dadin.

Implementazioa

Atal honetan, diseinatutako dialogo-sistema implementatuko da, modu iteratiboan.

Emangarria	Entregatze-data
Kodea	2022 / VI / 26
Memoria	2022 / VI / 26
Aurkezpena	2022 / VII / 01

A.2 Taula: Emangarrien entregatze datak

Ebaluazio protokoloa definitzea

Sortutako chatbot-aren funtzionamendua nolakoa den eta proba garaian azaldu daitezkeen arazoak ikusteko, ebaluazio protokolo bat definitu behar da. Horrela, dialogo-sistema probatuko duten erabiltzaileek protokolo hau jarraitu beharko dute, ondoren hauek sortutako datuak esanguratsuak izan daitezten.

Ebaluazioa eta emaitzak

Atal honetan, erabiltzaileen lagin batekin chatbot-aren funtzionamendua ebaluatuko da. Honen ostean, lortutako emaitzen analisi sakon bat egingo da, ondoren hobekuntza posibleak proposatu ahal izateko.

A.2.3 Emangarriak

Proiektu honen amaieran ondorengo emangarriak sortuko dira:

- Kodea
- Memoria
- Aurkezpena

A.2.4 Mugarriak

[A.2](#) Taulak sortu behar diren emangarrien entregatze-datak zehazten ditu.

A.2.5 *Gantt* diagrama

[A.2](#) Irudian proiektuaren garapenean jarraitu den *Gantt* diagrama ikusi daiteke. Bertan ataza bakoitzaren hasiera eta amaitze datak zehazten dira.

Lan-paketea		2022						
		Urtarrila	Otsaila	Martxoa	Apirila	Maiatza	Ekaina	Uztaila
Kudeaketa	Plangintza							
	Jarraipena eta kontrola							
	Memoria eta aurkezpena							
Oinarrizko teoria								
Diseinua eta inplementazioa	Sistemaren diseinua							
	Inplementazioa							
Ebaluazioa	Ebaluazio protokoloa definitzea							
	Ebaluazioa eta emaitzak							

A.2 Irudia: Gantt diagrama

A.3 Lan metodologia

Proiektua garatzeko ikaslearen ordenagailu pertsonala erabiliko da, honen baliabideekin nahikoa baita garapenerako erabiliko diren tresnak exekutatzeko.

A.3.1 Bilerak

Proiektuaren jarraipena egin ahal izateko, astero tutoreen eta ikaslearen arteko bilera bat egingo da, zehazki, asteartean 10:00etan. Biltzeko arazoren bat egonez gero, aldez aurretik email bidez beharrezkoak diren aldaketak zehaztuko dira. Bileretan, aurreko astean egindako lanaren laburpena eta hurrengo asterako helburuak finkatuko dira.

A.3.2 Planifikatutako ordutegiak

Ikasleak ez du aurredefinitutako ordutegirik jarraituko, baina errutina bat mantentzen saiatuko da.

A.4 Bideragarritasuna

Proiektuaren ataza nagusia finkatzeaz gain, beharrezkoa da proiektuaren bideragarritasuna bermatzea, hau aurrera eraman ahal izateko. Horretarako, ondorengo puntuak definituko dira:

- Baliabideen kostua: Proiektuaren garapenerako erabili diren tresna guztien kostua doakoa da eta hauek eskuragarri daude.
- Baliabideen funtzionamendu zuzena: Erabiltzen diren tresnen funtzionamendua zuzena da eta garapenaren zehar eskuragarri daude.
- Denbora: Proiektu aurrera eramateko nahikoa denbora dagoela ziurtatu behar da, azken momentuko desbiderapenak egon ez daitezen.
- Ikasleen eta tutoreen arteko komunikazioa: Komunikazio hau zuzeneko eta eraginkorra izan behar da, zalantzak eta mezuak ahalik eta azkarren transmititu ahal izateko.

A.5 Arriskuak eta prebentzioak

Proiektuaren garapenaren zehar, oso ohikoa da arriskuak eta ustekabeakoak gertatzea, eta hauek proiektuaren plangintza osoa aldatu dezakete. Hori dela eta, oso garrantzitsua da arrisku potentziala aldeztatik identifikatuta izatea, eta hauei aurre egiteko plan bat izatea. Bestalde, kontutan izan behar da, aldeztatik ikusi ez den arrisku bat ere gertatu daitekeela.

A.5.1 Arriskuak

- Orduen esleipena zuzena ez izatea: Ataza batek esleitutako ordu kopurua baino ordu gehiago behar izatea gertatu daiteke. Honek beste ataza batzuei ordu gutxiago ematea, eta ondorioz ataza horietan gutxiago sakontzea ekar dezake.
- Informazioa galtzea: Erabilitako ordenagailuak matxuraren bat izan dezake, eta ondorioz egindako lan guztia galtzeko arriskua dago.
- Garapenerako baliabide nahikorik ez izatea: Sortu den dialogo-sistema ez dago pentsatuta produktu komertzial bat izateko, eta ondorioz ez ditu erabiltzaile asko aldi berean zerbitzatuko. Hala ere, dialogo-sistemaren zerbitzaria ikaslearen ordenagailuan exekutatzeko denez, honek eragina izan dezake chatbot-aren funtzionamenduan.

A.5.2 Prebentzioa

Aipatutako arriskuei aurre egiteko, ondorengo prebentzio plana sortu da:

- Ataza bakoitzari esleitutako ordu kopuruarekin arazorik egon ez dadin, orduak kalkulatzeko garaian desbiderapenak egoteko arriskua kontutan hartuko da. Horrela, ataza bakoitzari behar dituen baino ordu gehiago esleituko zaizkio, modu horretan plangintza malguago bat sortuz.
- Informazioa ez galtzeko, egindako aldaketa guztiak *Github* plataformako errepositorio batean gordeko dira. Horrela, zerbait gertatuz gero, lan guztia bertan gordeko da, eta lanean jarraitzeko errepositoriotik kodea deskargatzearekin nahikoa izango da.
- Ikaslearen ordenagailuarekin dialogo-sistema exekutatzeko nahiko baliabide ez badaude, IXA taldearen zerbitzari bat erabiliko da.

B. ERANSKINA

Ebaluazio galdetegia

5 bilaketa desberdin egin dituzu, zenbatetan lortu dituzu albiste esanguratsuak? *
0 1 2 3 4 5
<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>
Itzulitako albisteak alde batera utzita, izan ditudan elkarrizketa guztietan chatbot-ak zentzuzko gauzak esan ditu? *
1 2 3 4 5
Ez nago batere ados <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> Guztiz ados
Aurkitu nahi nituen albisteak berriako web orrian baino azkarrago aurkitu ditut? *
1 2 3 4 5
Ez nago batere ados <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> Guztiz ados
Chatbot hau erabiliko nuke egunerokotasunean albisteak bilatzeko? *
1 2 3 4 5
Ez nago batere ados <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> Guztiz ados
Arazoren bat izan baduzu, edo hobekuntzaren bat baldin baduzu, hemen adierazi
Tu respuesta _____

B.1 Irudia: Ebaluaziorako erabili den galdetegia

C. ERANSKINA

Elkarrizketen adibideak

C.1 Intent sailkapen zuzena duen elkarrizketa

USER: Kaixo

CHATBOT: Kaixo! Ni Juliet chatbot-a naiz.

CHATBOT: Berria egunkariko artikulua modu errazago batean bilatzeko programatu naute. Bilaketak egiteko, "hitz_gakoak albisteak" moduko formatua erabili, mesedez. Adibidez: "Futboleko albisteak". Zalantzak badituzu, "laguntza" idatzi.

USER: Laguntza

CHATBOT: Albisteak edo kategoriak bilatzeko ondorengo formatuak onartzen dira:

- "Hitz_gakoak albisteak"
- "Hitz_gakoak berriak"
- "Hitz_gakoak notiziak"

Behin notiziak aurkitutakoan horrela aukeratu daitezke:

- Lehenengo, bigarrena edo hirugarrena.
- Bat, bi edo hiru.
- 1, 2 edo 3.
- Botoiak erabiliz.

USER: Futbol berriak

CHATBOT: 1) Futbolari eskertuta doa

2) Lehenbiziko aldiz emakumezko arbitroak epaile arituko dira gizonen Munduko Kopan

3) «Gogobeteta» esango du agur

C.2 Intent sailkapen okerra duen elkarrizketa

USER: Kaixo

CHATBOT: Kaixo! Ni Juliet chatbot-a naiz.

CHATBOT: Berria egunkariko artikuluak modu errazago batean bilatzeko programatu naute. Bilaketak egiteko, "hitz_gakoak albisteak" moduko formatua erabili, mesedez. Adibidez: "Futboleko albisteak". Zalantzak badituzu, "laguntza" idatzi.

USER: Errusia

CHATBOT: Kaixo! Ni Juliet chatbot-a naiz.

CHATBOT: Berria egunkariko artikuluak modu errazago batean bilatzeko programatu naute. Bilaketak egiteko, "hitz_gakoak albisteak" moduko formatua erabili, mesedez. Adibidez: "Futboleko albisteak". Zalantzak badituzu, "laguntza" idatzi.

USER: Errusia albisteak

CHATBOT: 1) Stoltenbergen esanetan, Putinek «hutsegite handi bat» egin du

2) Errusiako petrolioaren enbargoa blokeatzen jarraitzen du Hungariak

3) Ekialderako bidean

C.3 Bilaketa zuzena duen elkarrizketa

USER: Gizarteak notiziak

CHATBOT: 1) «Galdeketa irabazteko beharra»

2) Kontzertuetan eraso sexistak eragozteko protokolo bat ondu dute

3) EZKERRA, AGINTEA ESKURATZEKOTAN

USER: 1

CHATBOT: Eskoziako gizarte mugimenduek gaur ekingo diote independentziaren aldeko kanpainari. Erreferenduma noiz egingo den bainoago, independentzia erdiesteko arrazoiak eztabaidatu nahi dituzte.

<https://www.berria.eus/paperekoa/1898/018/001/2022-05-29/galdeketa-irabazteko-beharra.htm>

C.4 Bilaketa okerra duen elkarrizketa

USER: Zarauzko albistekak

CHATBOT: 1) Atzapar bete literatura

2) Literatura eskolan, liburuetatik aterata

USER: Lehenengoa

CHATBOT: Gaur amaituko da Literaturia, Arantxa Iturberen, Miren Narbaizaren eta Eider Rodriguezen 'Elkarrizketa baten soinua eta notak' ikuskizunaren emanaldi batekin. Goizean, liburu azoka eta aurkezpenak izango dira. Inoiz baino argitaletxe gehiago elkartu dira Zarautzen, udaberri giroan.

<https://www.berria.eus/paperekoa/1981/030/001/2022-05-29/atzapar-bete-literatura.htm>

USER: Zarautz albistekak

CHATBOT: Ez dut erantzunik aurkitu zure galderarentzat.

Bibliografia

- [Agerri et al., 2014] Agerri, R., Bermudez, J., and Rigau, G. (2014). IXA pipeline: Efficient and ready to use multilingual NLP tools. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3823–3828, Reykjavik, Iceland. European Language Resources Association (ELRA).
- [Bocklisch et al., 2017] Bocklisch, T., Faulkner, J., Pawlowski, N., and Nichol, A. (2017). Rasa: Open source language understanding and dialogue management.
- [Brandtzaeg and Følstad, 2017] Brandtzaeg, P. B. and Følstad, A. (2017). Why people use chatbots. In Kompatsiaris, I., Cave, J., Satsiou, A., Carle, G., Passani, A., Kontopoulos, E., Diplaris, S., and McMillan, D., editors, *Internet Science*, pages 377–392, Cham. Springer International Publishing.
- [Bunk et al., 2020] Bunk, T., Varshneya, D., Vlasov, V., and Nichol, A. (2020). Diet: Lightweight language understanding for dialogue systems.
- [Chakravarthy, 2020] Chakravarthy, S. (2020). Tokenization for natural language processing.
- [Honnibal and Montani, 2017] Honnibal, M. and Montani, I. (2017). spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.
- [Karpukhin et al., 2020] Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., and Yih, W.-t. (2020). Dense passage retrieval for open-domain question answering.
- [Khyani et al., 2021] Khyani, D., Siddhartha, B., Niveditha, N., and Divya, B. (2021). An interpretation of lemmatization and stemming in natural language processing. *Journal of University of Shanghai for Science and Technology*.

- [Luo et al., 2022] Luo, B., Lau, R. Y. K., Li, C., and Si, Y.-W. (2022). A critical review of state-of-the-art chatbot designs and applications. *WIREs Data Mining and Knowledge Discovery*, 12(1):e1434.
- [Màrquez et al., 2000] Màrquez, L., Padró, L., and Rodríguez, H. (2000). A machine learning approach to pos tagging. *Machine Learning*, 39(1):59–91.
- [Palit, 2020] Palit, S. (2020). Build a chatbot using rasa.
- [Pérez, 2020] Pérez, E. (2020). Qué es el efecto eliza, o por qué nos sorprende tanto leer un artículo "escrito" por una inteligencia artificial como gpt-3.
- [Robertson and Zaragoza, 2009] Robertson, S. and Zaragoza, H. (2009). The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3:333–389.
- [Santorini, 1990] Santorini, B. (1990). Part-of-speech tagging guidelines for the penn treebank project.
- [Sennrich et al., 2015] Sennrich, R., Haddow, B., and Birch, A. (2015). Neural machine translation of rare words with subword units.
- [Singhal et al., 2001] Singhal, A. et al. (2001). Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, 24(4):35–43.
- [Sinha et al., 2010] Sinha, G., Shahi, R., and Shankar, M. (2010). Human computer interaction. In *2010 3rd International Conference on Emerging Trends in Engineering and Technology*, pages 1–4.
- [Van Rossum and Drake, 2009] Van Rossum, G. and Drake, F. L. (2009). *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA.
- [Vlasov et al., 2019] Vlasov, V., Mosig, J. E. M., and Nichol, A. (2019). Dialogue transformers.
- [yeung Shum et al., 2018] yeung Shum, H., dong He, X., and Li, D. (2018). From eliza to XiaoIce: challenges and opportunities with social chatbots. *Frontiers of Information Technology & Electronic Engineering*, 19(1):10–26.
- [Zhao, 2017] Zhao, B. (2017). Web scraping. *Encyclopedia of big data*, pages 1–3.