

ŠIAULIŲ UNIVERSITETAS

Regionų plėtros institutas

Paulius Jacinkevičius

**NLP sistemų analizės programinės įrangos
sudarymas ir testavimas**

Bakalauro baigiamasis darbas

Vadovas: Doc. Dr. Asta Slotkienė

Šiauliai, 2020

TURINYS

1.	ĮVADAS	3
1.1	NLP sistemų pritaikymas „chatbot“ sistemoms	4
1.2	Klasterizacija.....	4
2.	REIKALAVIMŲ SPECIFIKACIJA.....	6
2.1	Sistemos paskirtis.....	6
2.2	Projekto dalyviai	6
2.3	Vartotojai	6
2.4	Įpareigojantys apribojimai.	6
2.5	Veiklos kontekstas	7
2.6	Produkto veiklos sfera.....	8
2.7	Nefunkciniai reikalavimai.....	10
2.8	Egzistuojantys sprendimai	12
2.9	Galimos sistemos kūrimo rizikos.....	12
2.10	Vartotojo dokumentacija ir apmokymas	12
2.11	Perspektyviniai reikalavimai	12
3.	TYRIMO METODOLOGIJA.....	13
3.1	Dokumento paskirtis	13
3.2	Tyrimo tikslai.....	13
3.3	Tyrimo apribojimai	13
3.4	Tyrimo techninė ir programinė įranga	13
3.5	Tyrimo metodika.....	14
3.6	Tyrimo struktūros diagrama:.....	19
3.7	Kokybė.....	19
4.	TESTAVIMAS	20
4.1	Testavimo paskirtis	20
4.2	Sistemos/PĮ funkcijų testavimas	21
4.3	Sistemos/PĮ charakteristikų testavimas	25
4.4	Diegimo testavimas.....	26
4.5	Testavimo išvados.....	26
5.	VARTOTOJO DOKUMENTACIJA.....	27
5.1	Vartotojo vadovo paskirtis.....	27
5.2	Sistemos funkcinis aprašymas.....	27
5.3	Sistemos vartotojo vadovas.....	27
5.4	Sistemos diegimo vadovas	28
6.	TYRIMO REZULTATAI.....	29
6.1	Tikslo klasifikacijos analizė.....	29
6.2	Sentimento analizės analizės rezultatai	30
7.	IŠVADOS.....	31
	LITERATŪRA.....	32
	TERMINŲ IR SANTRUMPŲ ŽODYNAS.....	33

1. ĮVADAS

Šiais laikais rinkoje yra daugybė įmonių, siūlančių natūraliosios kalbos apdorojimo sistemas kaip paslaugą, kuriomis naudojantis galima lengvai kurti sudėtingus projektus, susijusius su dirbtiniu intelektu. Įmonės, kuriančios automatizuotų pokalbių kambarių (angl. chatbot) sistemas įvairiems tikslams, dažnai negali apseiti be tokia paslaugą teikiančių produktų. Savarankiškai kurti savo tikslo klasifikavimo bei sentimentų analizavimo uždavinius spendžiančią sistemą savo projektui yra daug resursų ir investicijų reikalaujantis procesas. Didžiausias iššūkis šiandien tokioms įmonėms – išsirinkti kurią kalbos apdorojimo sistemą naudoti. Šis sprendimas gali priklausyti nuo kelių aspektų: kalba, teksto šaltinio rūšis, gramatinių klaidų taisymo galimybės, įvairių paslaugų pasiūla, bei sistemų pajėgumas ir kokybė. Tačiau internete nėra įrankių ir yra labai mažai atliktų tyrimų, kurie reguliariai tikrintų visų, nuolat tobulėjančių, sistemų charakteristikas. Tai neleidžia įmonių techniniams vadovams išsirinkti tiksliausiai, efektyviausiai ir greičiausiai veikiančią natūraliosios kalbos apdorojimo sistemą savo projektui. Skirtingi vartotojai reikalauja skirtingų paslaugų, tačiau visi reikalauja operacijos kokybės.

Šio darbo tikslas – sukurti programinę įrangą, galinčią automatiškai ištestuoti įvairių natūraliosios kalbos apdorojimo sistemų charakteristikas. Rezultatas – *tikslo klasifikacijos, sentimentų analizės* bei kitų charakteristinių užduočių paslaugų kokybės įvertinimas *klasterių analize* – suteikia galimybę įvairias natūraliosios kalbos apdorojimo sistemas palyginti tarpusavyje.

1.1 NLP sistemų pritaikymas „chatbot“ sistemoms

Chatbot sistema – tai automatizuota pokalbių kambario sistema. Remiantis dirbtinio intelekto galimybėmis, sistema savo vartotojui suteikia grįžtamąjį ryšį automatiškai: atsako į klausimus, reaguoja į komentarus, renka reikiamą informaciją, atlieka užduotis pagal vartotojo reikalavimus ir daug daugiau. Norint užtikrinti sklandų pokalbį tarp vartotojo ir chatbot sistemos, sistema privaloma tiksliai suprasti vartotojo parašytą tekstą (galbūt turintį gramatinių, skyrybos ar sintaksės klaidų, naudojanti įvairius žargonus ar išsireiškimus). Šiai paskirčiai yra atsakinga natūraliosios kalbos apdorojimo sistema.

Natūraliosios kalbos apdorojimas (angl. Natural Language Processing, arba NLP) – dirbtinio intelekto šaka, susijusi su natūraliosios (žmonių) kalbos interpretacija informacinių technologijų bei kompiuterinių programų pritaikymo poreikiams. Discipliną sudaro įvairūs uždaviniai: teksto skaitymas, automatinis kalbos vertimas, automatinis teksto taisymas, informacijos paieška, automatinis tekstų kūrimas ir skaitomo teksto atpažinimas. Chatbot sistemos dažniausiai eksploatuoja skaitomo teksto atpažinimo uždavinius, konkrečiau: tikslo analizės ir sentimentų analizės uždavinius.

Tikslo analizė (angl. intent analysis) – teksto tikslo ar prasmės aptikimo užduotis. Išanalizavus duotą tekstą, nustatoma apie ką yra rašoma.

Sentimentų analizė (angl. sentiment analysis) – teksto emocinės būsenos aptikimo užduotis. Išanalizavus duotą tekstą, nustatoma ar jis yra teigiamas (džiaugsmingas), neigiamas (pyktybiškas) ar neutralus (faktiškas, be emocijų užuominų).

1.2 Klasterizacija

Klasterizacija (arba klasterių analizė) yra tam tikrų objektų rinkinių (klasterių) grupavimo užduotis. Dviejų klasterių įvertinimas (arba klasterių validacija) yra vykdomas pasitelkiant tam tikrus algoritmus, atsižvelgiant į tam tikrus kriterijaus tipus.

Visi klasterių analės algoritmai yra paremti klasteriais. Klasteriai gali būti priskiriami keturioms reikšmėms – teisingas teigiamas, teisingas neigiamas, neteisingas teigiamas ir neteisingas neigiamas. Šios reikšmės priklauso nuo tikėtino ir gauto atsakymo santykio.

Lentelė 1: Klasterių grupavimo taisyklės

		Teisingas ataskymas	Atsakyta teisingai
TP	Teisingas teigiamas	Taip	Taip
TN	Teisingas neigiamas	Taip	Ne
FP	Neteisingas teigiamas	Ne	Taip
FN	Neteisingas neigiamas	Ne	Ne

Su NLP sistemų charakteristikos įvertinimu susiję duomenys yra neklasifikuoti – neturi iš anksto eksperto suklasifikuotų objektų, kuriuos galima būtų naudoti kaip standartą, ar etaloną. Toks kriterijus yra apibrėžtas klasterių validacijos *išorinio įvertinimo* algoritmuose. Vienas iš šių algoritmų – *F-matas* – yra skirtas .

Glaudumas (tikslumas) – tai teisingų ir visų gautų rezultatų proporcija, įvertinanti svarbių rezultatų procentą rinkinyje.

Išgava (jautrumas) – tai visų reikšmingų rezultatų proporcija, įvertinanti teisingai atsakytų svarbių rezultatų procentą rinkinyje.

F-matas – glaudumo ir išgavos harmoninis vidurkis, apibrėžiantis testo tikslumą.

$$P = \frac{TP}{TP + FP}$$

$$R = \frac{TP}{TP + FN}$$

, kur **P** – glaudumas ir **R** – išgava. F-matą galima apskaičiuoti:

$$F = 2 \cdot \frac{P \cdot R}{P + R}$$

2. REIKALAVIMŲ SPECIFIKACIJA

2.1 Sistemos paskirtis

2.1.1 Projekto kūrimo pagrindas

Dešimtys įmonių siūlo NLP paslaugas kaip produktą, tačiau šios paslaugos skiriasi savo kokybe, o būdų jas palyginti tarpusavyje galimybių nėra. Projekto kūrimo pagrindas – NLP sistemų gausa ir sistemų palyginimo galimybių stoka. Šio projekto rezultatai bus panaudoti chatbot kūrėjų įmonėje Feedyou.

2.1.2 Sistemos tikslai (paskirtis)

Testavimo sistema atliks testavimą bet kokioms NLP sistemoms naudojant REST API sąsają. Testavimo procesas apima spartos, glaudumo, išgavos ir f-mato dydžių matavimą.

2.2 Projekto dalyviai

2.2.1 Užsakovas

Jan Dvořák, Feedyou S.R.O., jan.dvorak@feedyou.agency, Čekijos Respublika.

2.2.2 Vykdytojas

Paulius Jacinkevičius, Feedyou S.R.O., paulius.jacinkevicius@feedyou.agency, Lietuva.

2.3 Vartotojai

Lentelė 2. Testuotojas

Vartotojo kategorija	Testuotojas
Vartotojo sprendžiami uždaviniai:	Įveda reikiamą NLP sistemos sąsajos informaciją, pateikia testavimo duomenis
Patirtis dalykinėje srityje:	Srities specialistas
Patirtis informacinėse technologijose:	Naujokas
Papildomos vartotojo charakteristikos:	Išmanyti kaip veikia NLP sistemos
Apsimokymo poreikis:	Reikia
Amžiaus grupė:	15-90

2.4 Įpareigojantys apribojimai.

2.4.1 Apribojimai sprendimui

Testavimo sistema buvo kurta JavaScript programavimo kalba, NodeJS platforma, pagerinant programavimo kokybę naudojantis TypeScript įrankiu. Ši konfigūracija buvo pasirinkta dėl jos galimybės vartoti įvairiose platformose, patogių įrankių (pvz. Request (<https://npmjs.com/package/request>)) ir patogumo kurti lengvasvoriais programas.

2. 4. 2 Diegimo aplinka

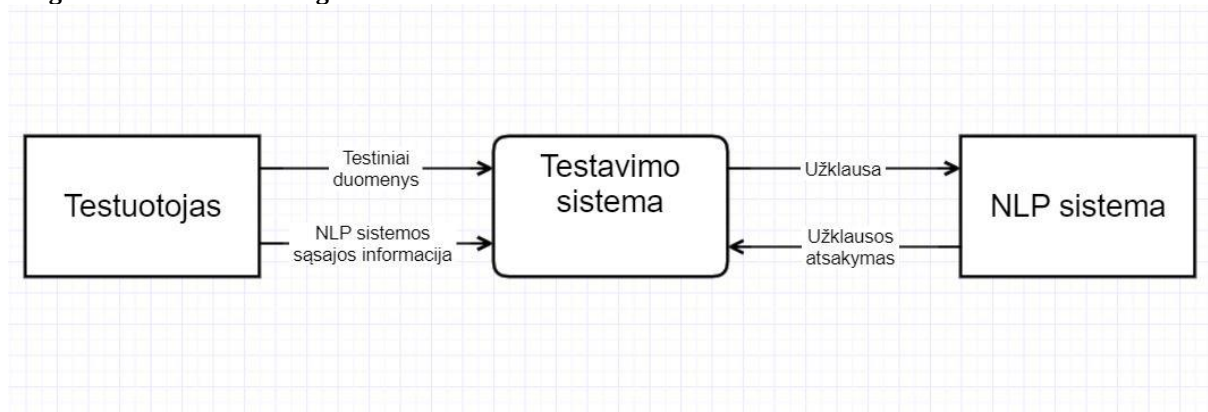
Testavimo sistemą palaikanti aplinka gali būti asmeninis kompiuteris arba viešos prieigos serveris.

2. 4. 3 Bendradarbiaujančios sistemos

PĮ bendradarbiauja su trečiosios šalies NLP sistemomis, naudojant REST API sąsają: ParallelDots, LUIS, Wit, DialogFlow, Watson, Geneea, Alquist.

2. 5 Veiklos kontekstas

Diagrama 1. Konteksto diagrama



2. 5. 1 Veiklos padalinimas

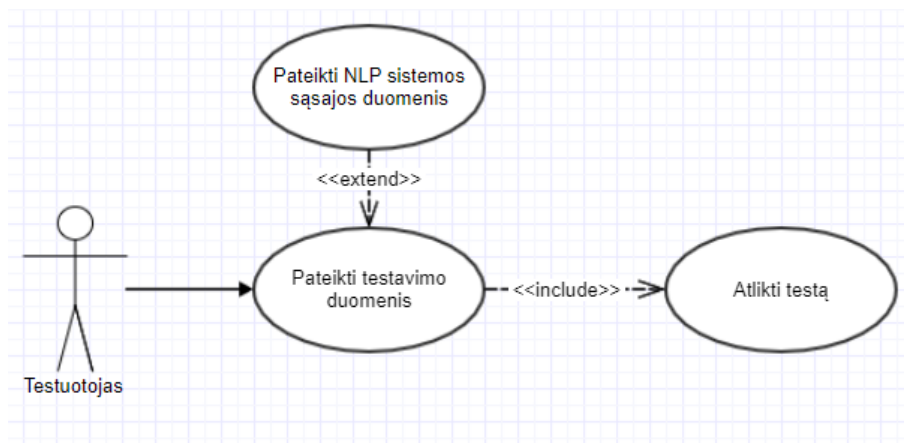
Lentelė 3. Veiklos padalinimas

Eil. Nr.	Įvykio pavadinimas	Įeinantys/išeinantys srautai
1.	NLP sistemų sąsajos informacija	Testuotojo nurodomas sąsajos URI adresas, kuriuo bus siunčiama užklausa su testiniais duomenimis.
2.	Testiniai duomenys	Testuotojo nurodomi testiniai duomenys, kuriais bus siunčiamos užklausos NLP sistemai ir jos tikrinami rezultatai.
3.	Užklausa	Testavimo sistemos siunčiama užklausa per REST API sąsają nurodytu URI adresu su nurodytais testiniais duomenimis.
4.	Užklausos atsakymas	NLP sistemos atsakymas su teksto analizės duomenimis, kurie bus tikrinami ir analizuojami testavimo sistemos.

2. 6 Produkto veiklos sfera

2. 6. 1 Sistemos ribos

Diagrama 2. Testuotojo panaudos atvejis



2. 6. 2 Panaudojimo atvejų sąrašas

Lentelė 4. Panaudojimo atvejo „Pateikti testavimo duomenis“ detalizavimas

Panaudojimo atvejis	Pateikti testavimo duomenis
Aprašymas	Testuotojas įveda testavimo duomenis duotu formatu.
Prieš-salyga	Atidaryti langą su įvesties forma.
Pagrindinis scenarijus	Įvedami testavimo duomenys arba paliekami numatytieji. Paspaudžiamas mygtukas „Send“.
Po-sąlyga	Vygdama analizė ir testuotojui parodomas testo rezultatas.

Lentelė 5. Panaudojimo atvejo „Pateikti NLP sistemos sąsajos duomenis“ detalizavimas

Panaudojimo atvejis	Pateikti NLP sistemos sąsajos duomenis
Aprašymas	Testuotojas įveda NLP sistemos sąsajos duomenis.
Prieš-salyga	Atidaryti langą su įvesties forma.
Pagrindinis scenarijus	Įvedami NLP sistemos sąsajos duomenys arba paliekami numatytieji. Paspaudžiamas mygtukas „Send“.
Po-sąlyga	Vygdama analizė ir testuotojui parodomas testo rezultatas.

Lentelė 6. Panaudojimo atvejo „Atlikti testą“ detalizavimas

Panaudojimo atvejis	Atlikti testą
Aprašymas	Testuotojui parodomi testavimo (analizės) rezultatai.
Prieš-salyga	Įvesti arba palikti numatytieji testavimo bei NLP sistemos sąsajos duomenys.
Pagrindinis scenarijus	Paspaudžiamas mygtukas „Send“ ir parodomi testavimo (analizės) rezultatai ekrane.
Po-sąlyga	Rezultatai pavaizduoti ekrane.

2. 6. 3 Funkciniai reikalavimai

Lentelė 7. Funkcinio reikalavimo “Užklausiama NLP sistema” detalizavimas

Reikalavimas #:	1	Reikalavimo tipas:	F	Panaudojimo atvejis nr.:	1
Aprašymas:	Užklausiama NLP sistema pagal nurodytą URI adresą ir testavimo duomenis.				
Pagrindimas:	Testuotojui norint ištestuoti NLP sistemą, reikia jai išsiųsti užklausą pagal tos sistemos URI sąsajos adresą ir duomenimis ir tada matuoti rezultato kokybę.				
Šaltinis:	Užsakovas				
Tikimo kriterijus:	NLP sistema sąveikauja, ir gražina atsaką JSON formatu.				
Užsakovo tenkinimas:	5/5				

Lentelė 8. Funkcinio reikalavimo “Matuojamas NLP sistemos tikslumas” detalizavimas

Reikalavimas #:	2	Reikalavimo tipas:	F	Panaudojimo atvejis nr.:	1
Aprašymas:	Matuojamas NLP sistemos tikslumas.				
Pagrindimas:	Norint palyginti NLP sistemas, reikia matematiškai apskaičiuoti jų tikslumo koeficientą.				
Šaltinis:	Užsakovas				
Tikimo kriterijus:	Atliekamos matematinės funkcijos				
Užsakovo tenkinimas:	5/5				

Lentelė 9. Funkcinio reikalavimo “Parodomas rezultatas” detalizavimas

Reikalavimas #:	3	Reikalavimo tipas:	F	Panaudojimo atvejis nr.:	1
Aprašymas:	Analizės rezultatas pavaizduojamas ekrane.				
Pagrindimas:	Norint daryti išvadas apie analizės rezultatus, reikia juos išvsti ekrane.				
Šaltinis:	Užsakovas				
Tikimo kriterijus:	Atvaizduojami testavimo rezultatai ekrane.				
Užsakovo tenkinimas:	2/5				

Lentelė 10. Funkcinio reikalavimo „Numatytieji testiniai duomenys“ detalizavimas

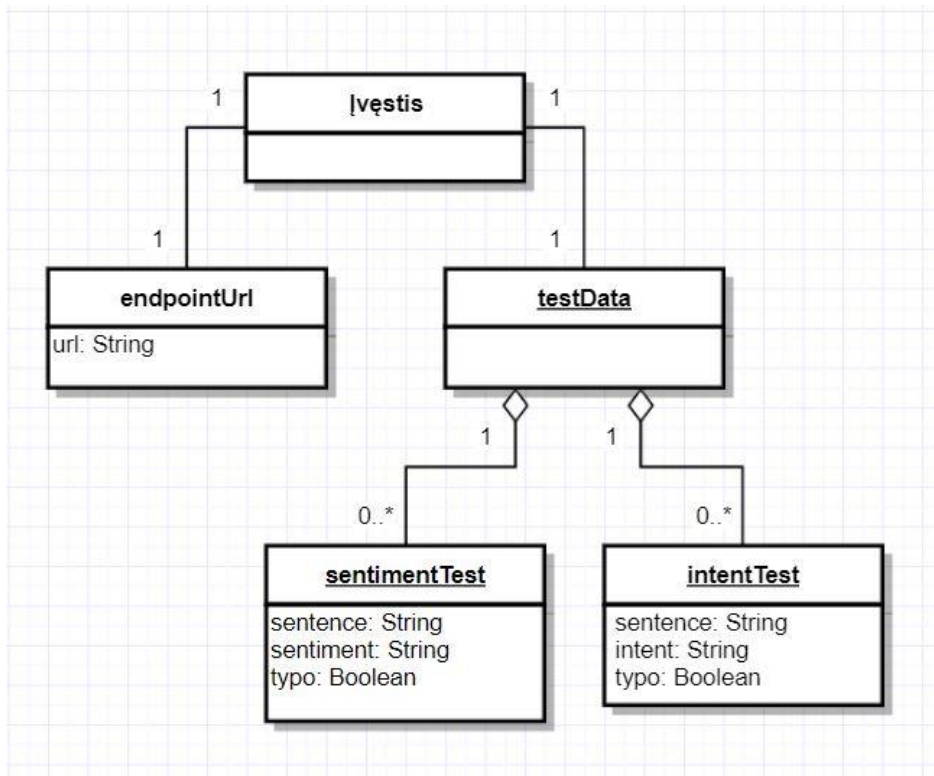
Reikalavimas #:	4	Reikalavimo tipas:	F	Panaudojimo atvejis nr.:	1
Aprašymas:	Galimybė naudoti numatytus testinius duomenis nenurodžius jų įvesties lange.				
Pagrindimas:	Norint greitai gauti rezultatą, kaip sistemos veikimo įrodymą, reikia patogaus būdo kaip kuo greičiau pradėti analizės procesą.				
Šaltinis:	Užsakovas				
Tikimo kriterijus:	Nenurodžius testinių duomenų įvesties lange, naudojami numatytieji duomenys, saugojami JSON faile, programos resursuose.				
Užsakovo tenkinimas:	2/5				

Lentelė 11. Funkcinio reikalavimo „Numatytasis URI adresas“ detalizavimas

Reikalavimas #:	5	Reikalavimo tipas:	F	Panaudojimo atvejis nr.:	1
Aprašymas:	Galimybė naudoti numatytus sąsajos adresus nenurodžius jų įvesties lange.				
Pagrindimas:	Norint greitai gauti rezultatą, kaip sistemos veikimo įrodymą, reikia patogaus būdo kaip kuo greičiau pradėti analizės procesą.				
Šaltinis:	Užsakovas				
Tikimo kriterijus:	Nenurodžius URI adreso įvesties lange, naudojamas numatytasis LUIS sistemos sąsajos URI adresas, saugojamas kaip viešasis kintamasis programoje.				
Užsakovo tenkinimas:	2/5				

2. 6. 4 Reikalavimai duomenims

Diagrama 3. Įvesties esybių ryšių diagrama



2. 7 Nefunkciniai reikalavimai

2. 7. 1 Reikalavimai sistemos išvaizdai

Lentelė 12. Nefunkcinio reikalavimo „Paprasta GUI“ detalizavimas

Reikalavimas #:	6	Reikalavimo tipas:	NF
Aprašymas:	Paprasta, paskirtą atliekanti vartotojo sąsaja.		
Pagrindimas:	Grafinė vartotojo sąsajos išbaigtumas yra nereikšmingas moksliniam tyrimui atlikti.		
Šaltinis:	Užsakovas		
Tikimo kriterijus:	GUI teikia įskaitomą sąsaja testuotojui.		
Užsakovo tenkinimas:	2/5	Užsakovo netenkinimas:	1/5
Priklausomybės:	-	Konfliktai:	Nėra

2. 7. 2 Reikalavimai panaudojamumui

Lentelė 13. Nefunkcinio reikalavimo „Naudojama anglų kalba“ detalizavimas

Reikalavimas #:	7	Reikalavimo tipas:	NF
Aprašymas:	Naudojama anglų kalba.		
Pagrindimas:	Sistema skirta anglų kalbą suprantantiems vartotojams.		
Šaltinis:	Užsakovas		
Tikimo kriterijus:	Vartotojo sąsaja pateikiama anglų kalba.		
Užsakovo tenkinimas:	4/5	Užsakovo netenkinimas:	3/5
Priklausomybės:	-	Konfliktai:	Nėra

2. 7. 3 Reikalavimai vykdymo charakteristikoms

Lentelė 14. Nefunkcinio reikalavimo „Pakartojimas” detalizavimas

Reikalavimas #:	8	Reikalavimo tipas:	NF
Aprašymas:	Galimybė pakartoti analizę naudojant tuos pačius duomenis.		
Pagrindimas:	Sistema turi tiksliai pakartoti operaciją pakartotinai be rezultato pokyčių, išskyrus NLP sistemos rezultato pokytį ir operacijos laiko rodiklį.		
Šaltinis:	Užsakovas		
Tikimo kriterijus:	Pakartojus operaciją, rezultatas pateikiamas su maža paklaida.		
Užsakovo tenkinimas:	3/5	Užsakovo netenkinimas:	3/5
Priklausomybės:	-	Konfliktai:	Nėra

2. 7. 4 Reikalavimai saugumui

Sistema yra pagalbinis įrankis, kuris neturi duomenų bazės su slaptais duomenimis ar kokios nors prieigos prie jautrių resursų. Dėl to saugumo principai yra neprivalomi, t.y. pakankami sklandžiam veikimui.

2.8 Egzistuojantys sprendimai

2.8.1 Galimos naudoti sistemos

Lentelė 15. Analogų palyginimas

Funkcija \ Sistema	Snips-AI	BotFuel	NLP analyzer (šis projektas)
Nemokama	+	+	+
Atviro kodo	-	-	+
Lankstumas (testuoja bet kokią sistemą)	-	-	+
Chatbot-orientuota	+	+	+
Testuoja Intent Classification	-	-	+
Testuoja Sentiment Analysis	-	-	+
Modifikuojama	-	-	+
Matuoja greitį	+	+	+
Matuoja Intent Classification tikslumą	+	+	+
Matuoja Sentiment Analysis tikslumą	-	-	+
Detalus analizės rezultatas	+/-	+	+
Vartotojams pasiekiamas įrankis	-	-	+
Detali dokumentacija	-	+/-	+

Nuorodos:

- Snips-AI: <https://github.com/snipsco/nlu-benchmark>
- BotFuel: <https://github.com/Botfuel/benchmark-nlp-2018>

2.9 Galimos sistemos kūrimo rizikos

Lentelė 16. Galimos sistemos kūrimo rizikos

Rizikos faktorius	Tikimybinis įvertinimas
Reikalavimų specifikacijos pasikeitimai realizavimo fazėje	10
Papildomų funkcijų prireikimas.	8

2.10 Vartotojo dokumentacija ir apmokymas

Vartotojas turi būti apmokytas specialisto. Vartotojui pateikiamas testuotojo vadovas, kuriame bus paaiškinta kaip naudotis programa, ką reiškia išvestyje vaizduojami duomenys, koku būdu buvo atlikta analize ir kaip modifikuoti sistemą tam prisireikūs.

2.11 Perspektyviniai reikalavimai

- Ši sistema privalo turėti galimybę ištestuoti bet kokią NLP sistemą, naudojančią REST API sąsają.
- Testavimo sistema gali testuoti NLP sistemas, kurios dar tik bus sukurtos ateityje.
- Turi veikti įvairiose platformose.
- Sistema turi turėti galimybę ateityje būti patalpina hosting'o serveryje.

3. TYRIMO METODOLOGIJA

3.1 Dokumento paskirtis

Šio tyrimo paskirtis – išanalizuoti NLP sistemų atsakų į užklausas rezultato tikslumą jų palyginimui ir taikomosios srities išsiaiškinimui.

3.2 Tyrimo tikslai

- Atlikti *sentimento analizės* analizę, kuri parodo sistemos veikimo efektyvumą atpažįstant sentimentą.
- Atlikti *tikslo klasifikacijos* analizę, kuri parodo sistemos veikimo efektyvumą klasifikuojant pokalbio tikslą.
- Atlikti bendrą sistemos veikimo spartos analizę, kuri parodo sistemos veikimo greitį.
- Palyginti populiariausias NLP sistemas ir išrinkti sistemą, kuri yra efektyviausia *sentimento analizės* tikslui.
- Palyginti populiariausias NLP sistemas ir išrinkti sistemą, kuri yra efektyviausia *tikslo klasifikacijos* tikslui.

3.3 Tyrimo apribojimai

Šiame tyrime dalyvavusios NLP sistemos buvo registruotos nemokamo išbandymo (angl. free trial) režimu. Tarp Luis teikiamų nemokamų paslaugų nėra profesionalaus apsimokymo paslaugos, bei apribota „Bing Spell Check“ klaidų tikrinimo apribota. Taip pat, Luis riboja maksimalų užklausų per minutę skaičių. Šie apribojimai šiek tiek įtakoja šios NLP sistemos analizės rezultatus.

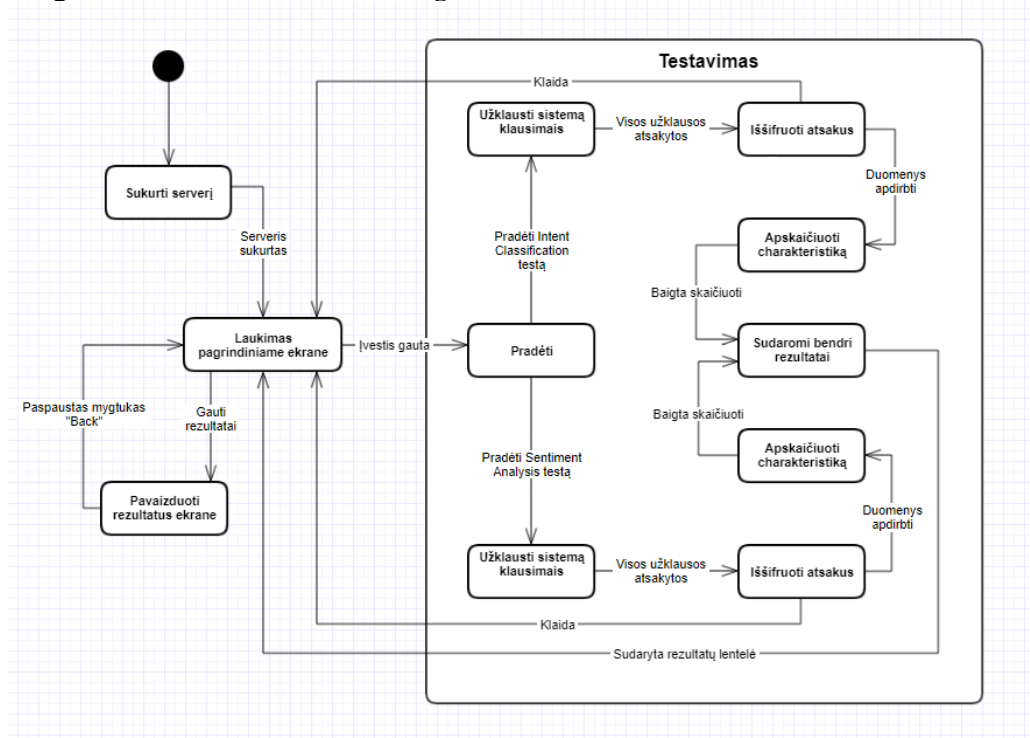
3.4 Tyrimo techninė ir programinė įranga

Priemonė	Paskirtis
Asmeninis kompiuteris	Skirtas paleisti ir įrengti vietinio vartotojo sąsaja su programa naudojant https://localhost:3000 adresą. Alternatyva: galima naudoti hosting paslaugas, kurios programą leis serveryje ir pasiekama statiniu IP adresu, pasiekiamą domenu.
Interneto prieiga	Skirta įrengti programos aplinką ir naudotis NLP sistemų sąsaja.
Node.js ir NPM bibliotekos	Skirtos suorganizuoti programos paleidimą ir sąsają tarp programoje vartojamų bibliotekų.
Visual Studio Code	Skirtas parašyti programinį kodą. Galimos įvairios alternatyvios IDE programos.

3.5 Tyrimo metodika

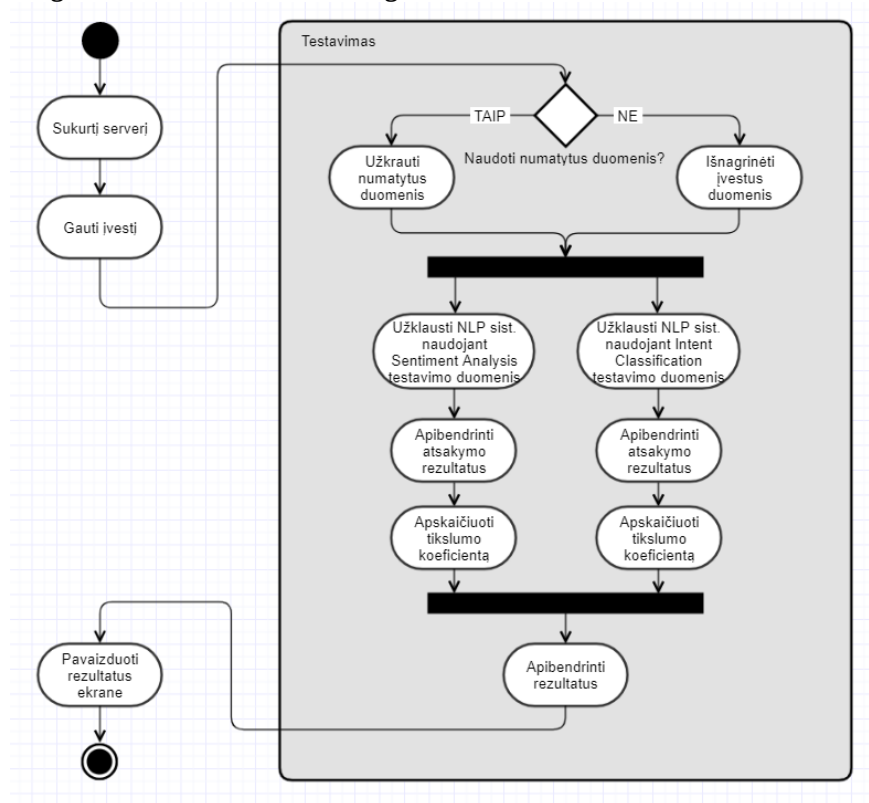
3.5.1 Bendroji tyrimo eiga

Diagrama 4. Sistemos būsenos diagrama.



Sistema pradeda darbą sukurs serverį su adresu localhost:3000. Vartotojas, prisijungęs šiuo adresu savo naršyklėje mato pagrindinį langą ekrane ir gali suvesti norimus duomenis, sistema tuo metu laukia. Sistemai gavus įvestį pradedamas testavimas. Testavimas yra sudarytas iš dviejų lygiagrečių ir simetriškų kelių: tikslo klasifikacijos analizavimas ir sentimentų analizės analizavimas. Jie prasideda užklausančiu nurodymu NLP sistemą nurodytais testiniais klausimais ir laukiama, kol visos užklausos atsakomos. Tada iššifruojami atsakai į subendrintą formatą ir, išskylus klaidas, nutraukiami visi procesai ir grįžtama į pagrindinį ekraną, arba, jeigu klaidų nerasta, analizė tęsiama. Toliau apskaičiuojama sistemos charakteristika ir sudaromi bendri rezultatai, čia lygiagretieji keliai susikerta. Bendri rezultatai siunčiami į pagrindinę sąsają, kurioje gauti rezultatai pavaizduojami ekrane. Paspaudus mygtuką „back“, grįžtama į pagrindinį įvesties langą.

Diagrama 5. Sistemos veiklos diagrama.



Sistemos veikla aprašoma pradedant nuo serverio sukūrimo. Sukūrus serverį, laukiama kol iš vartotojo bus gauta įvestis. Tada prasideda testavimo veikla, kuri yra sudaryta iš trijų dalių. Iš pradžių yra nusprendžiama, kokie testiniai duomenys bus naudojami – įvestieji ar numatytieji. Jeigu įvestieji, tai duomenys yra išnagrinėjami, o jeigu numatytieji, tai duomenys yra užkraunami iš failo. Toliau lygiagrečiai (asinchroniškai) vyksta du simetriški procesai: sentimentų analizė ir tikslo klasifikacija. Jie prasideda išsiuntus NLP sistemoms užklausas naudojant atitinkamus testinius duomenis (taip pat asinchroniškai) ir sulaukus visų atsakų, rezultatai apibendrinami ir apskaičiuojami tikslumo koeficientai abiem procesams. Finale rezultatai dar kartą apibendrinami ir pavaizduojami ekrane.

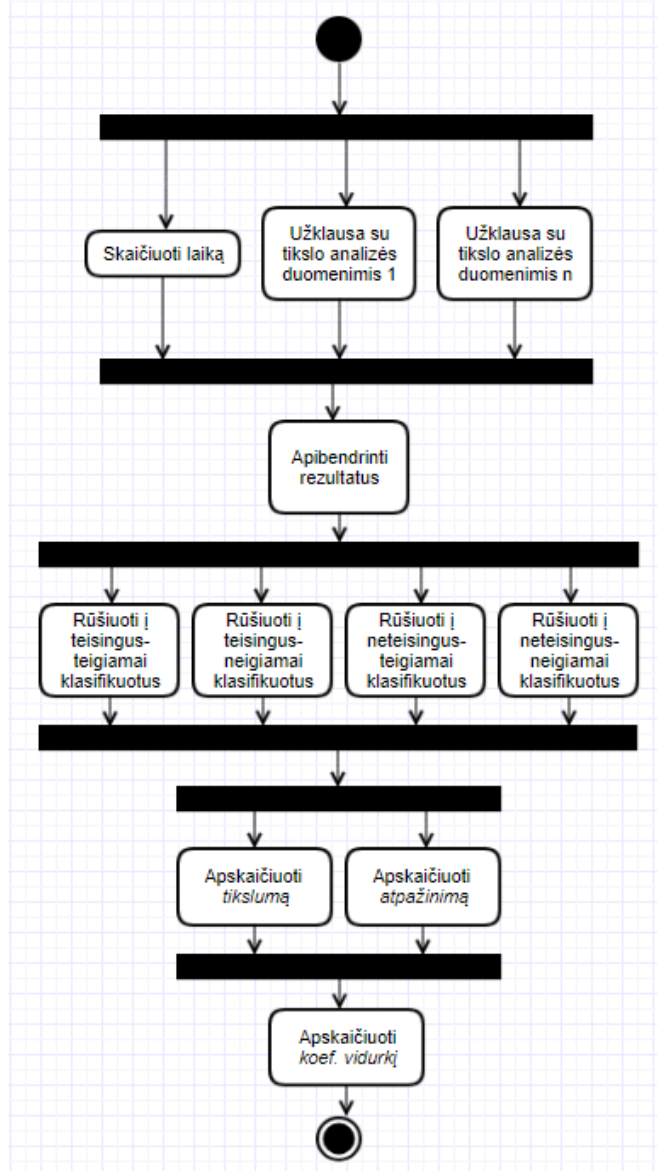
3. 5. 2 Tyrimo kriterijai

Lentelė 17. Tyrimo kriterijai.

Tyrimo pavadinimas	Tyrimo kriterijai
Tikslo klasifikacijos tikslumas	Apskaičiuoti n užklausų rezultatų F-matą
Tikslo klasifikacijos glaudumas	Apskaičiuoti n užklausų rezultatų glaudumo vertę
Tikslo klasifikacijos išgava	Apskaičiuoti n užklausų rezultatų išgavos vertę
Tikslo klasifikacijos užklausų trukmė	Apskaičiuoti kiekvienos iš n užklausų min/max trukmę milisekundėmis ir rasti jų vidurkį
Sentimento analizės tikslumas	Apskaičiuoti n užklausų rezultatų F-matą
Sentimento analizės glaudumas	Apskaičiuoti n užklausų rezultatų glaudumo vertę
Sentimento analizės išgava	Apskaičiuoti n užklausų rezultatų išgavos vertę
Sentimento analizės užklausų trukmė	Apskaičiuoti kiekvienos iš n užklausų min/max trukmę milisekundėmis ir rasti jų vidurkį

3. 5. 3 Tikslų klasifikacijos tyrimo kriterijaus eiga

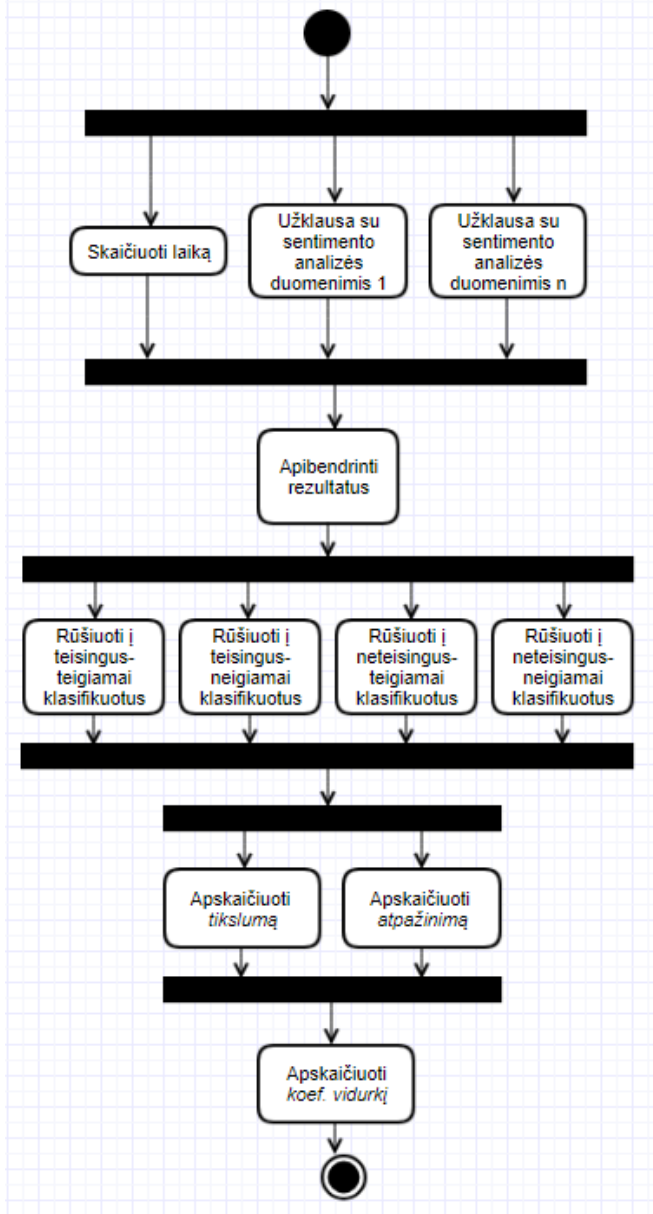
Diagrama 6. Tikslų klasifikacijos tyrimo veiklos diagrama.



Tikslų klasifikacijos analizė pradedama asinchroniškai uždavus užklausas su visais *Intent Classification* testiniais duomenimis (kiek duomenų, tiek užklausų) į vieną NLP sistemą. Sulaukus visų atsakų iš NLP sistemos, nuskaitomas užklausos trukmės laikas ir visų atsakų rezultatai yra apibendrinami į paprastesnį formatą, kadangi įvairios NLP sistemos siunčia skirtingo formato atsaką. Tada vyksta rezultato įvertinimo žingsnis, kuriame atsako rezultatai suskirstomi į teisingus-teigiamai, teisingus-neigiamai, neteisingus-teigiamai ir neteisingus-neigiamai klasifikuotus atvejus. Viesiems atvejams įvertinus apskaičiuojami tikslumas (angl. *precision*), atpažinimo vertė (angl. *recall*) ir bendras šių reikšmių vidurkio koeficientas (angl. *F-Measure*). Šios trys reikšmės apibūdina NLP sistemos tikslumą.

3. 5. 4 Sentimento analizės tyrimo kriterijaus eiga

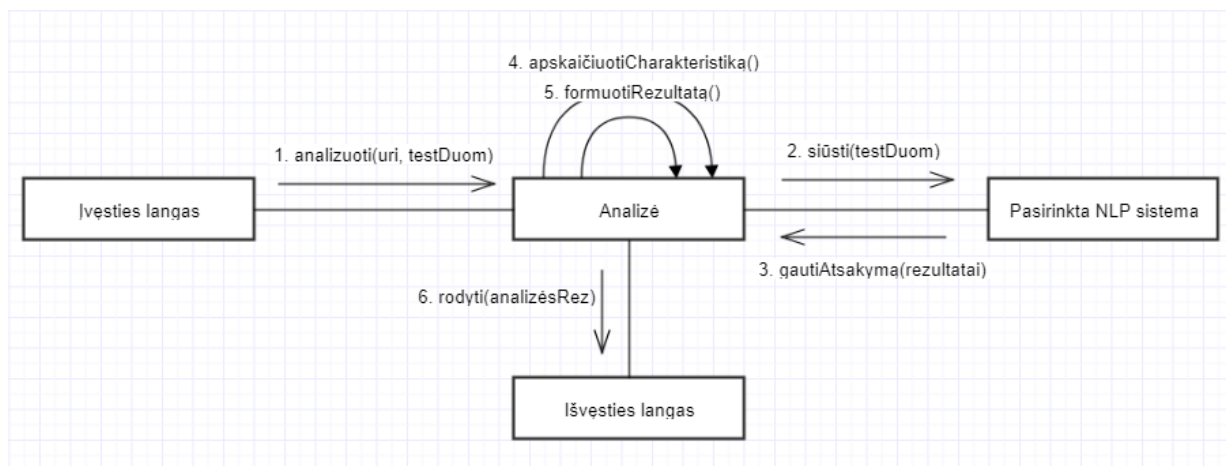
Diagrama 7. Sentimento analizės tyrimo veiklos diagrama.



Sentimento analizės tyrimas pradedamas asinchroniškai uždavus užklausas su visais *Sentiment Analysis* testiniais duomenimis (kiek duomenų, tiek užklausių) į vieną NLP sistemą. Sulaukus visų atsakų iš NLP sistemos, nuskaitomas užklauskos trukmės laikas ir visų atsakų rezultatai yra apibendrinami į paprastesnį formatą, kadangi įvairios NLP sistemos siunčia skirtingo formato atsaką. Tada vyksta rezultato įvertinimo žingsnis, kuriame atsako rezultatai suskirstomi į teisingus-teigiamai, teisingus-neigiamai, neteisingus-teigiamai ir neteisingus-neigiamai klasifikuotus atvejus. Visiems atvejams įvertinus apskaičiuojami tikslumas (angl. *precision*), atpažinimo vertė (angl. *recall*) ir bendras šių reikšmių vidurkio koeficientas (angl. *F-Measure*). Šios trys reikšmės apibūdina NLP sistemos tikslumą.

3.6 Tyrimo struktūros diagrama:

Diagrama 8. Tyrimo esybių ryšių diagrama



Tyrimo struktūra sudaryta iš:

- Testuotojo, kuris įveda testinius duomenis ir NLP sistemos URI.
- Išvesties ekrano, kuriame bus pavaizduotas rezultatas.
- NLP paslaugos tiekėjo, kuris vygdo NLP funkcijas per REST API sąsają. Pvz.: Wit, Luis, Cogito, ParallelDots, ...
- NLP analizės sistemos, kurioje vyksta operacijų paskirstymas ir rezultato apskaičiavimas.

3.7 Kokybė

Šio tyrimo rezultatas bus panaudotas chatbot kūrėjų įmonėje Feedyou S.R.O., Čekijoje. Užsakovas pareiškė susidomėjimą geriausios NLP sistemos pasirinkimo galimybėmis. Jam yra aktualu greitu metu tobuliname chatbot variklyje išrinkti ir pritaikyti tinkamiausią NLP sistemą, kuri bus naudojama tūkstančiuose chatbotų kasdien, o informacijos internete šia tema yra mažai.

Taip pat tikimasi praskaidrinti žmonių supratimą apie NLP sistemų charakteristikas, kadangi informacijos internete yra mažai, o dalis jos - pasenusi. NLP tobulėja ne dienom, o valandom savarankiško mokymosi ir pastovaus tobulinimo dėka. Šis projektas leidžia pakartoti tyrimą visiems norintiems nemokamai su bet kokia NLP sistema ir bet kokiais testiniais duomenimis.

Tyrimas gali būti išplėstas tirti NLP sistemų Įvardintų Esybių Atpažinimo (angl. *NER - Named Entity Recognition*), Kalbos Dalies Suskirstymo (angl. *POS-tagging – Part Of Speech tagging*) ir detalią Emocijos analizės (angl. *Emotion*) charakteristikas, tačiau jos nėra projekto užsakovo poreikis.

Tyrimo rezultatas – matematiškai pagrįsta NLP sistemų charakteristikų apskaita, paremta atpažinimo teorija (angl. *pattern recognition*), taigi patikimumo kokybė yra aukšta.

4. TESTAVIMAS

4.1 Testavimo paskirtis

4.1.1 Testavimo tikslai

- Funkcinis testavimas – parodyti sistemos funkcijų tinkamą veikimą eksplotavimui;
- Scenarijų testavimas – patikrinti įvesties resurso klaidų žalos rezultatui apsaugą.
- Stresinis testavimas – stebėti kaip sistema veikia didelio apkrautumo sąlygomis;
- Suderinamumo testavimas – nustatyti programinės įrangos suderinamumą sistemai funkcionuoti.

4.1.2 Testavimo resursai

Testavimas atliktas naudojantis asmeniniu kompiuteriu su Windows 10 OS.

Naudota programinė įranga: Visual Studio Code, Luis.ai, Wit.ai.

Pagalbinė programinė įranga:

- Testų atvejams aprašyti: Jest (<https://jestjs.io/>).
- Stresinio testo įrankis: Artillery (<https://artillery.io/>).

CLI aplinka: NodeMon (<https://nodemon.io/>) ir Nodejs (<https://nodejs.org/>).

4.1.3 Pagrindiniai apribojimai

Testuojant bandoma kurti įvairius funkcijų panaudojimo scenarijus. Sistema, kurta kliento serverio pagrindu, todėl klientas turi turėti bet kokią naršyklę.

NLP sistemų analizės rezultatai gali turėti įtakos dėl nestabilaus ar lėto interneto greičio dėl to, kad skirtingos NLP sistemos reikalauja vartotojų turėti stabilų interneto greitį. NLP sistemai atmetus užklausą dėl per daug žemo interneto greičio, grąžinama klaida ir analizės procesas yra nutraukiamas, o to pasekmė – neatitinkantis lūkesčių rezultatas. Tokiu atveju testavimas turi būti atliktas dar kartą, su stabilesniu internetu.

NLP sistemų analizės rezultatai taip pat gali turėti įtakos dėl per didelio užklausų kiekio per tam tikrą laiką dėl to, kad NLP sistemos neleidžia vartotojams, kurie naudoja nemokamą prieigą prie sistemos, apkrauti sistemą dažnomis užklausomis. Užklausų dažnis yra nurodytas NLP sistemos platontiko internetinėje svetainėje. Šio testavimo atveju, kuriame naudojama sistema Luis, užklausų dažnis negali viršyti 50 užklausų per sekundę, o Wit – 1 užklausos per sekundę.

4.2 Sistemos/PĮ funkcijų testavimas

Testo metu buvo atliekamas sistemos realizuotų modulių ir funkcijų išsamus testavimas, kad įsitikinti ar nėra paliktų klaidų programiniame kode ir ar visos funkcijos veikia tinkamai ir tiksliai.

Lentelė 18. Funkcinis vidurkio apskaičiavimo testavimas

Testo tikslas:		Vidurkio apskaičiavimas				
Testo ID	Reikalavimai/tikslai	Įvykis/įvestis	Laukiamas rezultatas	Gautas rezultatas	T/N	Pastabos
1	Vidurkio apskaičiavimas standartiniu atveju	[1,2,3,4,5]	3	3	T	
2	Vidurkio apskaičiavimas naudojant sudėtingus realiuosius skaičius	[7.7765, 4.203, 12.0201, 6.1, 3.99, 7.9104]	7	7	T	JavaScript suapvalina kintamuosius
3	Vidurkio apskaičiavimas naudojant tuščią aibę (grėsmė dalybai iš nulio)	[]	0	0	T	Dalyba iš nulio išvengta sėkmingai
4	Vidurkio apskaičiavimas naudojant vieną skaičių	[0]	0	0	T	

Lentelė 19. Funkcinis santykio apskaičiavimo testavimas

Testo tikslas:		Santykio apskaičiavimas				
Testo ID	Reikalavimai/tikslai	Įvykis/įvestis	Laukiamas rezultatas	Gautas rezultatas	T/N	Pastabos
5	Santykio apskaičiavimas standartiniu atveju	4, 8	0.5	0.5	T	
6	Santykio apskaičiavimas, kai antrasis parametras yra 0 (grėsmė dalybai iš nulio)	12, 0	0	0	T	Dalyba iš nulio išvengta sėkmingai
7	Santykio apskaičiavimas, kai antrasis parametras yra mažesnis už pirmąjį	4, 2	2	2	T	

Lentelė 20. Funkcinis sąsajos adreso gavimo testavimas

Testo tikslas:		Sąsajos adreso gavimas pagal sarašinį (enum) kintamąjį				
Testo ID	Reikalavimai/tikslai	Įvykis/įvestis	Laukiamas rezultatas	Gautas rezultatas	T/N	Pastabos
8	Luis adreso gavimas pagal enum reikšmę	Api.Luis	Luis sąsajos adresas	Luis sąsajos adresas	T	
9	Wit adreso gavimas pagal enum reikšmę	Api.Wit	Wit sąsajos adresas	Wit sąsajos adresas	T	
10	Watson adreso gavimas pagal enum reikšmę	Api.Watson	Watson sąsajos adresas	Watson sąsajos adresas	T	
11	Adreso gavimas pagal bet kokią string reikšmę	“hello world”	“”	“”	T	
12	Adreso gavimas pagal undefined reikšmę	Undefined	“”	“”	T	

13	Wit adreso gavimas pagal string reikšmę	“Wit”	“”	“”	T	
14	Luis adreso gavimas pagal string reikšmę	“Luis”	“”	“”	T	
15	Watson adreso gavimas pagal string reikšmę	“Watson”	“”	“”	T	
16	Adreso gavimas pagal enum reikšmę, kai įvestas numatytasis sąsajos adresas	Api.Luis; bei įvestas nustatytasis sąsajos adresas	Nustatytas sąsajos adresas	Nustatytas sąsajos adresas	T	
17	Adreso gavimas pagal undefined reikšmę, kai įvestas numatytasis sąsajos adresas	undefined; bei įvestas nustatytasis sąsajos adresas	Nustatytas sąsajos adresas	Nustatytas sąsajos adresas	T	

Lentelė 21. Funkcinis tikslumo apskaičiavimo testavimas

Testo tikslas:		Tikslumo apskaičiavimas				
Testo ID	Reikalavimai/tikslai	Įvykis/įvestis	Laukiamas rezultatas	Gautas rezultatas	T/N	Pastabos
18	Tikslumo apskaičiavimas, kai parametru sudėtis lygi 0 (grėsmė dalybai iš nulio)	0, 0	0	0	T	Dalyba iš nulio išvengta sėkmingai
19	Tikslumo apskaičiavimas standartinių parametru atveju	10, 5	2/3	2/3	T	

Lentelė 22. Funkcinis atpažinimo apskaičiavimo testavimas

Testo tikslas:		Atpažinimo apskaičiavimas				
Testo ID	Reikalavimai/tikslai	Įvykis/įvestis	Laukiamas rezultatas	Gautas rezultatas	T/N	Pastabos
20	Recall apskaičiavimas, kai parametru suma lygi 0 (grėsmė dalybai iš nulio)	0, 0	0	0	T	Dalyba iš nulio išvengta sėkmingai
21	Recall apskaičiavimas standartinių parametru atveju	10, 5	2/3	2/3	T	

Lentelė 23. Funkcinis bendro tikslumo vidurkio apskaičiavimo testavimas

Testo tikslas:		Bendro tikslumo harmoninio vidurkio apskaičiavimas				
Testo ID	Reikalavimai/tikslai	Įvykis/įvestis	Laukiamas rezultatas	Gautas rezultatas	T/N	Pastabos
22	Harmoninio vidurkio apskaičiavimas, kai parametru suma lygi 0 (grėsmė dalybai iš nulio)	0, 0	0	0	T	Dalyba iš nulio išvengta sėkmingai
23	Harmoninio vidurkio apskaičiavimas standartinių parametru atveju	10, 5	2/3	2/3	T	

Lentelė 24. Funkcinis tikslo klasifikacijos testavimas

Testo tikslas:		Tikslo klasifikacija				
Testo ID	Reikalavimai/tikslai	Įvykis/įvestis	Laukiamas rezultatas	Gautas rezultatas	T/N	Pastabos
24	Standartinis analizės scenarijus Wit sąsajai	Standartiniai testiniai duomenys, Wit sąsaja	Pilnas JSON užklauso atsakas (reikšminiai duomenys)	Pilnas JSON užklauso atsakas (reikšminiai duomenys)	T	
25	Standartinis analizės scenarijus Luis sąsajai	Standartiniai testiniai duomenys, Luis sąsaja	Pilnas JSON užklauso atsakas (reikšminiai duomenys)	Pilnas JSON užklauso atsakas (reikšminiai duomenys)	T	
26	Neteisingų testinių duomenų analizės scenarijus	Neteisingo formato testiniai duomenys, Wit sąsaja	undefined	undefined	T	
27	Neegzistuojančio sąsajos adreso analizės scenarijus	Standartiniai testiniai duomenys, neegzistuojanti sąsaja	“no endpoint url defined for undefined”	“no endpoint url defined for undefined”	T	
28	Analizė su undefined duomenimis ir undefined sąsaja. Tikimasi gauti klaidos aprašymą.	Undefined, undefined	String tipo užklauso klaidos atsakas	String tipo užklauso klaidos atsakas	T	

Lentelė 25. Funkcinis sentiment analizės testavimas

Testo tikslas:		Sentimento analizė				
Testo ID	Reikalavimai/tikslai	Įvykis/įvestis	Laukiamas rezultatas	Gautas rezultatas	T/N	Pastabos
29	Standartinis analizės scenarijus Wit sąsajai	Standartiniai testiniai duomenys, Wit sąsaja	Pilnas JSON užklauso atsakas (reikšminiai duomenys)	Pilnas JSON užklauso atsakas (reikšminiai duomenys)	T	
30	Standartinis analizės scenarijus Luis sąsajai	Standartiniai testiniai duomenys, Luis sąsaja	Pilnas JSON užklauso atsakas (reikšminiai duomenys)	Pilnas JSON užklauso atsakas (reikšminiai duomenys)	T	
31	Neteisingų testinių duomenų analizės scenarijus	Neteisingo formato testiniai duomenys, Wit sąsaja	undefined	undefined	T	
32	Neegzistuojančio sąsajos adreso analizės scenarijus	Standartiniai testiniai duomenys, neegzistuojanti sąsaja	“no endpoint url defined for undefined”	“no endpoint url defined for undefined”	T	
33	Analizė su undefined duomenimis ir undefined sąsaja. Tikimasi gauti klaidos aprašymą.	Undefined, undefined	String tipo užklauso klaidos atsakas	String tipo užklauso klaidos atsakas	T	

Lentelė 26. Funkcinis bendros NLP sistemų analizės scenarijų testavimas

Testo tikslas:		Bendra NLP sistemų analizė				
Testo ID	Reikalavimai/tikslai	Įvykis/įvestis	Laukiamas rezultatas	Gautas rezultatas	T/N	Pastabos
34	Wit analizė su neteisingais duomenimis. Tikimasi gauti klaidos aprašymą.	Neteisingo formato testiniai duomenys, kai trūksta simbolio “\”; Wit sąsaja	String tipo klaidingos įvesties klaidos atsakas	String tipo klaidingos įvesties klaidos atsakas	T	“Failed parsing input data. Go back and check for errors: SyntaxError: Unexpected token } in JSON at position 147”
35	Wit analizė su neteisingais duomenimis. Tikimasi gauti klaidos aprašymą.	Neteisingo formato testiniai duomenys, kai trūksta vieno elemento; Wit sąsaja	String tipo klaidingos įvesties klaidos atsakas	String tipo klaidingos įvesties klaidos atsakas	T	“Failed parsing input data. Go back and check for errors: Error: Testing Data has bad format.”
36	Wit analizė su neteisingais duomenimis. Tikimasi gauti klaidos aprašymą.	Neteisingo formato testiniai duomenys, kai visi elementai neegzistuojantys; Wit sąsaja	String tipo klaidingos įvesties klaidos atsakas	String tipo klaidingos įvesties klaidos atsakas	T	“Failed parsing input data. Go back and check for errors: Error: Testing Data has bad format.”
37	Luis analizė su tuščiais duomenimis. Tikimasi gauti klaidos aprašymą.	Tušti testiniai duomenys; Luis sąsaja	String tipo klaidingos įvesties klaidos atsakas	String tipo klaidingos įvesties klaidos atsakas	T	“Failed parsing input data. Go back and check for errors: SyntaxError: Unexpected end of JSON input”
38	Analizė su standartiniais testiniais duomenimis neegzistuojančiu sąsajos adresu. Tikimasi gauti klaidos aprašymą.	Standartiniai testiniai duomenys; Neegzistuojanti sąsaja	String tipo klaidingos sąsajos klaidos atsakas	String tipo klaidingos sąsajos klaidos atsakas	T	“Invalid URI “www.15min.lt””
39	Wit analizė su standartiniais duomenimis	Standartiniai testiniai duomenys; Wit sąsaja	Pilnas JSON užklauso atsakas (reikšminiai duomenys)	Pilnas JSON užklauso atsakas (reikšminiai duomenys)	T	
40	Wit analizė su numatytais duomenimis	Numatyti testiniai duomenys; Wit sąsaja	Pilnas JSON užklauso atsakas (reikšminiai duomenys)	Pilnas JSON užklauso atsakas (reikšminiai duomenys)	T	
41	Luis analizė su standartiniais duomenimis	Standartiniai testiniai duomenys; Luis sąsaja	Pilnas JSON užklauso atsakas (reikšminiai duomenys)	Pilnas JSON užklauso atsakas (reikšminiai duomenys)	T	

42	Luis analizė su numatytais duomenimis	Numatyti testiniai duomenys; Luis sąsaja	Pilnas JSON užklauso atsakas (reikšminiai duomenys)	Pilnas JSON užklauso atsakas (reikšminiai duomenys)	T	
43	Luis analizė su neteisingais duomenimis. Tikimasi gauti klaidos aprašymą.	Neteisingo formato testiniai duomenys, kai trūksta vieno elemento; Luis sąsaja	String tipo klaidingos įvesties klaidos atsakas	String tipo klaidingos įvesties klaidos atsakas	T	“Failed parsing input data. Go back and check for errors: Error: Testing Data has bad format.”
44	Luis analizė su neteisingais duomenimis. Tikimasi gauti klaidos aprašymą.	Neteisingo formato testiniai duomenys, kai trūksta simbolio “]”; Luis sąsaja	String tipo klaidingos įvesties klaidos atsakas	String tipo klaidingos įvesties klaidos atsakas	T	“Failed parsing input data. Go back and check for errors: SyntaxError: Unexpected token } in JSON at position 147”
45	Analizė su undefined duomenimis ir undefined sąsaja. Tikimasi gauti klaidos aprašymą.	Undefined; undefined	String tipo klaidingos įvesties klaidos atsakas	String tipo klaidingos įvesties klaidos atsakas	T	“Cannot read property ‘IntentTesting’ of undefined”

4.3 Sistemos/PĮ charakteristikų testavimas

4.3.1 Našumo testavimas

Testavimas buvo atliktas naudojantis nemokamu NPM įrankiu – Artillery (<https://artillery.io/>).

Artillery našumo testavimas veikia siunčiant N simuliuotų vartotojų per sekundę ištiesas T sekundes.

Lentelė 27. Sistemos sąsajos našumo testavimas

Testo tikslas:		Našumo testavimas				
Testo ID	Reikalavimai /tikslai	Įvykis/įvestis	Laukiamas rezultatas	Gautas rezultatas	T/N	Pastabos
46	Paprastas sistemos apkrautumas simuliuojant vartotojus	5 virtualius vartotojus per sekundę, kurie naudotųsi pradinio langu. Testas vykdomas 60 sekundžių.	Sistema veikia be trikdžių, kiekvienas prisijungimas sėkmingas, sistemos procesorius neapkrautas.	Sistema veikė be trikdžių. 300 prisijungimų sėkmingi. Procesoriaus apkrovimas mažas: 3.8% - 4.2%. Atminties suvartojimas mažas: 68 MB	T	
47	Vidutinis sistemos apkrautumas simuliuojant vartotojus	100 virtualius vartotojus per sekundę, kurie naudotųsi pradinio langu. Testas vykdomas 60 sekundžių.	Sistema veikia be trikdžių, kiekvienas prisijungimas sėkmingas, sistemos procesorius neapkrautas.	Sistema veikė be trikdžių. 6000 sėkmingų prisijungimų. Procesoriaus apkrovimas didelis: 52%. Atminties suvartojimas vidutinis: 85 MB.	T	
48	Stiprus sistemos apkrautumas simuliuojant vartotojus	256 virtualius vartotojus per sekundę, kurie naudotųsi pradinio langu. Testas vykdomas 120 sekundžių.	Sistema veikia be trikdžių, kiekvienas prisijungimas sėkmingas, sistemos procesorius neapkrautas.	Sistema veikė labai prastai, 16340 iš 30721 prisijungimų sėkmingi. Procesoriaus apkrovimas buvo per didelis – nuo 55.5% iki 57%. Atminties suvartojimas vidutinis: 107 MB	N	Tokia sistemos prisijungimų apyvarta yra nereali pritaikymo atvejui

4.3.2 Suderinamumo testavimas

PĮ naudojamų priklausomybių (dependencies) versijos yra automatiškai surašomos package.json faile jų įrašymo metu.

Lentelė 28. Sistemos suderinamumo testavimas

Testo tikslas:						
Testo ID	Reikalavimai/tikslai	Įvykis/įvestis	Laukiamas rezultatas	Gautas rezultatas	T/N	Pastabos
49	request 2.88.0	Siunčiama HTTP užklausa NLP sistemai	Sėkmingas užklauskos atsakas	Sėkmingas užklauskos atsakas	T	
50	express 4.16.4	Sukuriamas serverio klientas 3000 kanalu	Sėkmingai sukurtas serveris ir localhost:3000 veikia teisingai	Sėkmingai sukurtas serveris ir localhost:3000 veikia teisingai	T	
51	node 8.11.4	Paleidžiama programa	Viskas veikia	Viskas veikia	T	

4.4 Diegimo testavimas

NodeJS moduliai yra suderinti taikymui įvairiose platformose.

Lentelė 29. Sistemos diegimo testavimas

Testo tikslas:		Patikrinti kaip programos veiks prie nurodytų TĮ ir PĮ				
Testo ID	Reikalavimai/tikslai	Įvykis/įvestis	Laukiamas rezultatas	Gautas rezultatas	T/N	Pastabos
52	Programos realizavimas Windows OS	Windows 10	Veikia	Veikia	T	
53	Programos realizavimas Windows OS	Windows 7	Veikia	Veikia	T	
54	Programos realizavimas Mac OS	Mac OS Sierra 10.12	Veikia	Veikia	T	
55	Programos realizavimas Linux OS	Ubuntu 9.1	Veikia	Veikia	T	

4.5 Testavimo išvados

Funkcinis sistemos testavimas parodė, kad visos funkcijos (2) veikia tiksliai, lanksčiai ir apie klaidas vartotojui praneša išsamiai.

Našumo testavimas buvo sėkmingas (46, 47, 48), nors ir PĮ pasirodė stipriai neveiksni esant tūkstančiams vartotojų vienu metu (48). Turint omenyje, jog visas programos tikslas yra atlikti testavimą, o ne naršyti tūkstančiams vartotojų, našumas, iš esmės, yra nereikšmingas. Express karkasas, ant kurio yra pagrįsta ši programinė įranga, atitinkamoje kompiuterinėje sistemoje (galingame serveryje) gali būti labai našus, o kaip nešiojamajame kompiuteryje, našumas buvo patenkinamas.

Suderinamumo testavime (55) matyti, kad PĮ gali veikti visose populiariosiose operacinėse sistemose.

5. VARTOTOJO DOKUMENTACIJA

5.1 Vartotojo vadovo paskirtis

Šis vartotojo vadovas aprašo sistemos funkcijas, kaip naudotis sistema ir diegimo instrukciją.

5.2 Sistemos funkcinis aprašymas

Sistema leidžia ištestuoti norimos NLP sistemos charakteristikas, suteikia įžvalgą į jos rezultato tendencijas.

5.3 Sistemos vartotojo vadovas

1. Sistemos paleidimas

- 1.1 Atidaromas norimas CLI terminalas (Command Prompt (Windows), Git Bash (Windows), MINGW (Windows, Linux, Mac OS), Linux CLI (Linux), ir kt.).
- 1.2 Terminalas naviguojamas į aplanką su programine įranga. Dažniausiai naudojama komanda: „*cd C:\adresas\į\direktoriją\sū\programine\įranga*“. Arba Windows operacinėje sistemoje galima iššaukti Git Bash terminalą tiesiai iš aplanko spaudžiant dešinyjį pelės klavišą aplanke su PĮ ir pasirinkus „Git Bash Here“. Alternatyva – atsidarius projektą su Visual Studio Code, integruotas terminalas su teisinga navigacija atidaromas automatiškai.
- 1.3 Vykdoma komanda „*npm start*“ ir spaudžiamas *Enter* mygtukas. Jeigu paleidimas buvo sėkmingas, terminale matomas užrašas „*Express listening on port 3000...*“.
- 1.4 Atidaroma bet kokia naršyklė ir vedamas adresas „*localhost:3000*“ ir spaudžiamas *Enter* mygtukas.

2. Duomenų įvestis

- 2.1 Sąsajoje egzistuoja 5 interakcijos (sąveikos). Jos yra (vardijama iš viršaus į apačią): teksto įvesties laukas, išskleidžiamas meniu, žymimasis langelis su užrašu „*Use my own training data*“, didelis teksto įvesties laukas su įrašytu jame tekstu ir slankijuoste dešinėje, ir mygtukas „*SEND*“.
- 2.2 Teksto įvesties lauke tikimasi nurodyti NLP sistemos sąsajos URI adresą. Jeigu adresas nenurodomas, išskleidžiamame meniu po juo galima pasirinkti numatytą sąsajos URI adresą prie vienos iš nurodytų NLP sistemų, kurios bus naudojamos analizėje.
- 2.3 Išskleidžiamas meniu leidžia pasirinkti numatytos NLP sistemos URI adresą. Norint pasirinkti vieną iš šių sistemų, paspauskite ant jos pavadinimo. Jeigu teksto įvesties laukas virš išskleidžiamo meniu yra užpildytas, tada ši funkcija neveiks.
- 2.4 Nepažymėjus žymimąjį langelį su užrašu „*Use my own training data*“, NLP analizės sistema naudos numatytuosius testinius duomenis, kurie yra saugojami projekto aplanke, **app/resources/testData.json** faile. O pažymėjus langelį, NLP analizės sistema naudos testinius duomenis tiesiai iš didelio teksto įvesties lauko po žymimoju langeliu.
- 2.5 Didelis teksto įvesties laukas virš mygtuko „*SEND*“ – testinių duomenų įvesties vieta. Nepažymėjus žymimojo langelio virš šio lauko, testiniai duomenys iš jo bus nenaudojami.

Šiame lange jau yra įrašytas pavyzdinis testiniai duomenys JSON formatu, kuriais galima naudotis vedant norimus testinius duomenis. Nenurodžius „IntentClassification“ arba „SentimentTesting“ elemento, atitinkamai, tikslo klasifikacijos arba sentimento analizės testai nebus vykdomi.

2. 6 Mygtukas *SEND* išsiunčia užklausas NLP sistemai pagal nurodytą (arba numatytą) sąsajos URI adresą su nurodytais (arba numatytais) testiniais duomenimis ir baigus analizę nukreipia puslapį į rezultatų apžvalgos puslapį.

3. Analizės rezultatų apžvalgos langas

3. 1 Šiame lange yra vienintelis mygtukas - „*BACK*“. Jį paspaudus grįžtama į pradinį langą adresu *localhost:3000*.
3. 2 Šiame lange matomi analizės rezultatai. Jie yra išdėstyti pagal kategorijas: „Intent Testing Results“ ir „Sentiment Testing Results“. Jeigu įvesti testiniai duomenys buvo neteisingo formato, arba atsirado kitokia klaida, vietoj analizės rezultatų bus rodoma klaidos aprašo žinutė. Kitu atveju, kiekvienoje analizės kategorijoje bus rodomi įvairūs rezultatų apie NLP sistemos analizę duomenys.

5. 4 Sistemos diegimo vadovas

1. Įdiegti WinRar (win-rar.com) ar kitą archyvų valdymo programą ir NodeJS (nodejs.org).
2. Išpakuojamas archyvas naudojant WinRar ar kita archyvų valdymo programa.
3. Išpakuoto archyvo kataloge atidaromas konsolės terminalas. Tinka „Command Prompt“, „GitBash“, „MINGW“ ar kitas „CLI“ įrankis veikiantis pasirinktoje operacinėje sistemoje.
 - 3.1. Command Prompt terminalo atidarymo instrukcija Windows:
Win→“cmd”→Enter→“cd [archyvo/direktorijos/adresas]”
 - 3.2. Git Bash terminalo atidarymo instrukcija Windows:
[Atidaryti archyvo katalogą]→[Dešinysis Pelės Klavišas direktorijoje]→Git Bash Here
4. Įvesti komandą „npm install“ ir spaudžiamas Enter.

6. TYRIMO REZULTATAI

6.1 Tikslo klasifikacijos analizė

Lentelė 30. Tikslo klasifikacijos analizės rezultatai

NLP sistema	Min/Max/Vid užklauskos trukmė (ms)	Klaidų sprendimas	Neteisingų neigiamų sk./visų	Neteisingų teigiamų sk./visų	Teisingų neigiamų sk./visų	Teisingų teigiamų sk./visų	Glaudumas (<i>Precision</i>)	Išgava (<i>Recall</i>)	F-matas
Geneea	290/360/659	0.5556	0.0795	0.2841	0.125	0.5114	0.6429	0.8654	0.7377
Alquist	171/1450/2164	0.5	0.2045	0.1705	0	0.625	0.7853	0.7534	0.7692
Luis	931/1069/1249	0.428	0	0.4483	0.2759	0.2759	0.381	1	0.5517
Wit	1121/1225/1322	0.9286	0.06897	0.0345	0.207	0.69	0.9524	0.909	0.9302

Tikslo analizės tyrime dalyvavo keturios NLP sistemos: Geneea, Alquist, Luis ir Wit. Visos šios sistemos turėjo vieną apmokymo ciklą, ir buvo užklaustos tokiomis pačiomis užklausomis.

- Geneea pasižymi sparčiausiu užklauskų greičiu, vidutiniškomis gramatinių klaidų sprendimo galimybėmis, ir surinko vidutinišką F-mato balą.
- Alquist (naudojama Amazon Echo) sparta yra nepastovi, o surinko F-mato balą vos didesnę už Geneea. Lyginant šias NLP sistemas kartu, Alquist glaudumo ir išgavos vertės yra panašios – tai yra optimalios, balansuotos NLP sistemos bruožas.
- Luis analizės metu pasirodė prasčiausiai tiek klaidų sprendimo, tiek glaudumo bei F-mato vertėse. Tačiau pasižymi geriausiu neteisingų-teigiamų užklauskų sprendimu. Ši savybė, dėja, nėra labai vertinga chatbot aplikacijoms.
- Wit (sukurta Facebook) geriausiai sprendė gramatinius klaidas ir surinko auksčiausią glaudimo, išgavos, bei F-mato balą.

Atsižvelgiant į įvairius NLP sistemos paslaugų poreikius Chatbot sistemoms, rekomenduoju Wit natūraliosios kalbos apdorojimo sistemą visoms Chatbot sistemoms, kurios ją naudotų tikslo atpažinimo uždaviniams spręsti. Jeigu užklauskos sparta yra svarbesnis aspektas, tada rekomenduočiau Geneea natūraliosios kalbos apdorojimo sistemą.

6.2 Sentimento analizės analizės rezultatai

Lentelė 31. Sentimento analizės analizės rezultatai

NLP sistema	Min/Max/Vid užklauskos trukmė (ms)	Klaidų sprendimas	Neteisingų neigiamų sk./visų	Neteisingų teigiamų sk./visų	Teisingų neigiamų sk./visų	Teisingų teigiamų sk./visų	Glaudumas (<i>Precision</i>)	Išgava (<i>Recall</i>)	F-matas
Geneea	-	-	-	-	-	-	-	-	-
Alquist	-	-	-	-	-	-	-	-	-
Luis	636/874/1294	0.4	0.1923	0.3846	0.1538	0.2692	0.4118	0.5833	0.4828
Wit	1081/1276/1426	0.6	0.1154	0.3462	0.2308	0.308	0.4706	0.7272	0.5713

Sentimento analizėje dalyvavo dvi NLP sistemos: Luis ir Wit. Geneea bei Alquist tokios paslaugos nesiuo. Abi sistemos turėjo vieną apmokymo ciklą, ir buvo užklaustos tokiomis pačiomis užklausomis.

- Luis surinko žemesnes vertes visuose aspektuose, išskyrus užklauskų apdirbimo greityje ir neteisingų-neigiamų užklauskų sprendime.
- Wit įvertinimas sentimentų analizės tyrime yra auščiausias daugumoje sričių.

Atsižvelgiant į sentimentų analizės analizės rezultatus, rekomenduoju tik Wit sentimentų analizės paslaugas Chatbot sistemoms.

7. IŠVADOS

Remiantis subjektyviu chatbot tikslų apibrėžimu, bei rinkos poreikio NLP sistemų testavimo įrankiams, 4-ame skyriuje buvo nustatyta reikalavimų analizei specifikacija. 5-ame skyriuje apibrėžta tyrimo metodologija, remiantis matematiniais sprendimais, pagrįstais 3-ajame skyriuje. Darbo eigoje atlikta tyrimo programos vientisumas, atlikti testavimai, aprašyti 6-ajame skyriuje. Galiausiai, apibendrinti tyrimo rezultatai ir pateiktos trumpos rekomendacijos renkant tarp kelių natūraliosios kalbos apdorojimo sistemų, 8-ame skyriuje. Šiame, paskutiniame, skyriuje pateikiamos darbo atlikėjo išvados ir pastebėjimai, apibendrinantys tyrimą, jo naudą, bei pritaikymą versle.

Šio tyrimo rezultatai buvo panaudoti ir vertingi įmonėje Feedyou S.R.O. Šio tyrimo rezultatų dėka, buvo lengvai išsirinkta efektyvi NLP sistema, kuri kasdien apdoroja tūkstančius užklausų iš šimtų žmonių. Įmonės techninis vadovas buvo labai patenkintas rezultatais. Papildomai, šio tyrimo rezultatai buvo naudingi mano pažįstamajam, Tomášui Vodrážkai, Belgijos KU Leuven universiteto, dirbtinio intelekto mokslų studento, savo baigiamojo magistro darbui, tema „NLP systems modelling analysis“.

Pačio tyrimo medžiagos nauda yra ginčijama, nes analizės konkurentai – subjektyviai pasirinkti iš populiarių konkurentų grupės. Pagrindinė šio darbo nauda – sukurtas viešai prieinamas įrankis, leidžiantis tirti įvairias NLP sistemas, kurios šiame tyrime nedalyvavo, bei naudoti savaip pritaikytus testinius duomenis. Manau, kad Chatbot kūrėjams šis įrankis neš didelę naudą ateinančius daugelį metų dėl savo lanksčių savybių.

Šis darbas nėra panacėja - keletas papildomų savybių ir patobulinimų galėtų išplėsti galimybių lauką – tačiau atsižvelgus į reikalavimų specifikaciją, savo tikslus praktikoje pasiekė.

LITERATŪRA

1. David M. W. Powers, Flinders University of South Australia. (2011). „*Evaluation: From Precision, Recall and F-Measure to ROOC, Informedness, Markedness & Correlation*“. [http://www.flinders.edu.au/science_engineering/fms/School-CSEM/publications/tech_reps-research_artfcts/TRRA_2007.pdf]
2. Howard W.R. (2007-02-20). „*The Pattern Recognition Basis of Artificial Intelligence*“. [<http://www.dontveter.com/basisofai/basisofai.html>]
3. Andreas Klintberg (2017-05-22). Meltwater darbuotojo tinklaraštis. „*Explaining precision and recall*“ [<https://medium.com/@klintcho/explaining-precision-and-recall-c770eb9c69e9>]
4. Andrew R. Freed (2016-10-31). IBM darbuotojo su Watson tinklaraštis. „*Cognitive system testing: Natural Language Processing unit testing*“. [<https://developer.ibm.com/watson/blog/2016/10/31/cognitive-system-testing-natural-language-processing-unit-testing/>]
5. Andrew R. Freed (2016-11-07). IBM darbuotojo su Watson tinklaraštis. „*Cognitive system testing: Overall system accuracy*“ [<https://developer.ibm.com/watson/blog/2016/11/07/cognitive-system-testing-part-5-overall-system-accuracy-testing/>]
6. Peters Florian, University of Liege, Belgija. (2018). Baigiamasis magistro darbas. „*Master thesis : Design and implementation of a chatbot in the context of customer support*“ [https://matheo.uliege.be/bitstream/2268.2/4625/6/Thesis_PETERS_Florian.pdf]
7. A. Pinto, H. G. Oliveira, A. O. Alves. University of Coimbra, Portugalija (2016-03). „*Comparing the Performance of Different NLP Toolkits in Formal and Social Media Text*“ [<http://drops.dagstuhl.de/opus/volltexte/2016/6008/pdf/OASICS-SLATE-2016-3.pdf>]
8. Daniel Braun, Technical University of Munich, Vokietija (2017-08-15). „*Evaluating Natural Language Understanding Services for conversational Question Answering Systems*“ [<http://workshop.colips.org/wochat/@sigdial2017/documents/SIGDIAL22.pdf>]
9. Raul Garreta (2015-08-13). MonkeyLearn darbuotojo tinklaraštis. „*Sentiment Analysis APIs Benchmark*“ [<https://monkeylearn.com/blog/sentiment-analysis-apis-benchmark/>]
10. Nguyen Trong Canh (2018-06-18). Botfuel darbuotojo tinklaraštis. „*Benchmarking Intent Classification services*“ [<https://medium.com/botfuel/benchmarking-intent-classification-services-june-2018-eb8684a1e55f>]

TERMINŲ IR SANTRUMPŲ ŽODYNAS

Terminas ar Santrupa	Santrupos vertimas angliškai	Santrupos vertimas lietuviškai	Reikšmė
NLP	Natural Language Processing	Natūraliosios Kalbos Apdorojimas	Disciplina, tirianti natūraliųjų (žmonių) kalbų naudojimo informacinių technologijų sritims galimybes.
NLP sistema			Įmonė, siūlanti NLP operacijas atliekančias paslaugas.
Sistema			Specifinių programų, užduočių, veiksmų ir duomenų visuma.
Hosting			Tinklapio talpinimo serveryje ir prieigos prie jo domeno pagalba paslauga.
Serveris			Specialus kompiuteris arba programa, skirta kitų kompiuterių aptarnavimui.
IDE	Integrated Development Environment	Integruota Kūrimo Aplinka	Programa, padedanti programuoti kitas programas.
URI	Uniform Resource Identifier	Universalusis Išteklių Identifikatorius	Simbolių eilutė, apibūdinanti resursą. Kitaip suprantama kaip tiklalapio adresas.
GUI	Graphical User Interface	Grafinė Vartotojo Sąsaja	Kompiuterio ekranas, kuriame grafiškai vaizduojama informacija.
CLI	Command-line Interface	Komandų Eilutės Sąsaja	Programa - terminalas, leidžiantis vygdyti komandomis valdomas programas.
PĮ	Software	Programinė Įranga	
Chatbot			Suprogramuota paslauga, kuri naudodamasi taisyklėmis ir dirbtiniu intelektu, atlieka jam priskirtas marketingo, komunikacijos, informacijos rinkimo ir kitas užduotis.
REST	Representational State Transfer		Architektūrinis stilius, skirtas interneto paslaugoms kurti gaunant ir manipuliuojant ištekliais
API	Application Programming Interface	Aplikacijų Programavimo Sąsaja	Sąsaja, kurią suteikia kompiuterinė sistema tam, kad kita viena programa galėtų apsieisti duomenimis su kita programa.
JSON	JavaScript Object Notation	JavaScript Objektų Žymėjimas	Formatas, perduodantis duomenų objektus, sudarytus iš atributo ir reikšmės porų