Kauno technologijos universitetas

Informatikos fakultetas

Mašininio mokymo taikymo klausimų kūrimui bei atsakymui lietuvių kalba tyrimas

Baigiamasis magistro projektas

Kristupas Talačka

Projekto autorius

Doc. Mantas Lukoševičius

Vadovas

Kaunas, 2025

Kauno technologijos universitetas

Informatikos fakultetas

Mašininio mokymo taikymo klausimų kūrimui bei atsakymui lietuvių kalba tyrimas

Baigiamasis magistro projektas

Programų sistemų inžinerija (6211BX011)

Kristupas Talačka

Projekto autorius

Doc. Mantas Lukoševičius

Vadovas

Dr. Šarūnas Packevičius

Recenzentas

Kauno technologijos universitetas

Informatikos fakultetas

Kristupas Talačka

Mašininio mokymo taikymo klausimų kūrimui bei atsakymui lietuvių kalba tyrimas

Akademinio sąžiningumo deklaracija

Patvirtinu, kad:

1. baigiamąjį projektą parengiau savarankiškai ir sąžiningai, nepažeisdama(s) kitų asmenų autoriaus ar kitų teisių, laikydamasi(s) Lietuvos Respublikos autorių teisių ir gretutinių teisių įstatymo nuostatų, Kauno technologijos universiteto (toliau – Universitetas) intelektinės nuosavybės valdymo ir perdavimo nuostatų bei Universiteto akademinės etikos kodekse nustatytų etikos reikalavimų;

2. baigiamajame projekte visi pateikti duomenys ir tyrimų rezultatai yra teisingi ir gauti teisėtai, nei viena šio projekto dalis nėra plagijuota nuo jokių spausdintinių ar elektroninių šaltinių, visos baigiamojo projekto tekste pateiktos citatos ir nuorodos yra nurodytos literatūros sąraše;

3. įstatymų nenumatytų piniginių sumų už baigiamąjį projektą ar jo dalis niekam nesu mokėjęs (-usi);

4. suprantu, kad išaiškėjus nesąžiningumo ar kitų asmenų teisių pažeidimo faktui, man bus taikomos akademinės nuobaudos pagal Universitete galiojančią tvarką ir būsiu pašalinta(s) iš Universiteto, o baigiamasis projektas gali būti pateiktas Akademinės etikos ir procedūrų kontrolieriaus tarnybai nagrinėjant galimą akademinės etikos pažeidimą.

Kristupas Talačka

Patvirtinta elektroniniu būdu

Kauno technologijos universitetas

Informatikos fakultetas

Baigiamojo magistro projekto užduotis

Projekto tema	Mašininio mokymo taikymo klausimų kūrimui bei atsakymui lietuvių kalba tyrimas
Reikalavimai ir sąlygos (tikslinti pavadinimą pagal poreikį)	
Vadovas / Vadovė	
	(vadovo pareigos, vardas, pavardė, parašas) (data)

Talačka Kristupas. Mašininio mokymo taikymo klausimų kūrimui bei atsakymui lietuvių kalba tyrimas. Magistro baigiamasis projektas / vadovas doc. Mantas Lukoševičius; Kauno technologijos universitetas, informatikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Programų sistemų inžinerija.

Reikšminiai žodžiai: Mašininis mokymasis, natūraliosios kalbos apdorojimas, tekstą generuojantys dirbtinio intelekto modeliai.

Kaunas, 2025. 57 p.

Santrauka

Šiame darbe aprašyta sistema, skirta automatiniam klausimų generavimui ir atsakymų pateikimui, naudojant natūralios kalbos apdorojimo (NLP) metodus lietuvių kalba. Sistema sukurta siekiant pagerinti mokymosi ir verslo procesus, leidžiant naudotojams generuoti klausimus bei atsakymus, pasitelkiant iš anksto ištreniruotus arba individualiai priderintus dirbtinio intelekto modelius.

Literatūros apžvalgoje analizuojami klausimų bei atsakymų generavimo metodai, esami sprendimai ir jų taikymas mažai išteklių turinčioms kalboms, tokioms kaip lietuvių. Pateikiamas modelių architektūrų, duomenų rinkinių ir vertinimo metrikų palyginimas.

Darbo tiriamojoje dalyje aprašomi duomenų rinkiniai, suformuoti naudojant automatizuotus duomenų praturtinimo metodus bei pateikiami modelių treniravimo ir vertinimo eksperimentai, kuriuose lyginami modeliai, treniruoti naudojant tik pradinius duomenis ir modeliai, treniruoti naudojant praturtintus duomenų rinkinius. Tyrimo rezultatai parodė, kad praturtinimo metodai ženkliai pagerina klausimų bei atsakymų generavimo modelių veikimo tikslumą. Geriausi rezultatai pasiekti taikant atvirkštinį vertimą.

Talačka Kristupas. A Study of the Application of Machine Learning to
Answering and Generating Questions in Lithuanian. Master's Final Degree Project / supervisor doc. Mantas Lukoševičius; Informatics faculty, Kaunas University of Technology.

Study field and area (study field group): Program system engineering.

Keywords: Machine learning, natural language processing, text-generating artificial intelligence models.

Kaunas, 2025. 57 p.

Summary

This project presents a system for automatic question generation and answering using natural language processing (NLP) methods in the Lithuanian language. The system is designed to support educational and business processes, allowing users to generate questions and answers using either pre-trained or custom fine-tuned artificial intelligence models.

The literature review analyzes existing methods for question generation and answering, their application to low-resource languages such as Lithuanian, and compares model architectures, datasets, and evaluation metrics.

The experimental part describes datasets created using automated data augmentation methods and presents training and evaluation experiments. The experiments compare models trained solely on the original data and models trained using augmented datasets. The results show that augmentation methods significantly improve question answering and question generation model performance. The best results were achieved using the back-translation technique.

Turinys

Lentelių sąrašas

Paveikslų sąrašas

Santrumpų ir terminų sąrašas

Santrumpos:

AI – Artificial Intelligence (dirbtinis intelektas). Kompiuterinių sistemų gebėjimas atlikti užduotis, kurios įprastai reikalauja žmogaus intelekto

D.I. – Dirbtinis intelektas. Lietuviškas dirbtinio intelekto sutrumpinimas.

CPU – Central Processing Unit (centrinis procesorius). Pagrindinis kompiuterio komponentas, atliekantis bendro pobūdžio skaičiavimus ir programų vykdymą.

GPU – Graphics Processing Unit (vaizdo plokštė). Specializuotas procesorius, skirtas grafikos apdorojimui ir plačiai naudojamas neuroninių tinklų mokymui dėl didelio lygiagretumo.

LLM – Large Language Model (didelis kalbos modelis). Didelės apimties neuroninis kalbos modelis, apmokytas su daug duomenų, gebantis generuoti, analizuoti ir interpretuoti tekstą.

NLP – Natural Language Processing (natūralios kalbos apdorojimas). Dirbtinio intelekto šaka, apimanti kalbos analizę, generavimą ir supratimą kompiuterinėmis priemonėmis.

QA – Question Answering (atsakymas į klausimus). NLP užduotis, kurios metu iš pateikto konteksto siekiama rasti atsakymą į konkretų klausimą.

QG – Question Generation (klausimų kūrimas). NLP užduotis, kurios metu iš pateikto teksto konteksto automatiškai sugeneruojami prasmingi klausimai.

RAM – Random Access Memory (operatyvioji atmintis). Laikinoji kompiuterio atmintis.

API – Application Programming Interface. Programavimo sąsaja, leidžianti skirtingoms sistemoms keistis duomenimis.

GDPR – General Data Protection Regulation. Bendrasis duomenų apsaugos reglamentas, nustatantis asmens duomenų tvarkymo principus Europos Sąjungoje.

HTTP – HyperText Transfer Protocol. Protokolas, naudojamas interneto svetainių ir API komunikacijai.

JSON – JavaScript Object Notation. Duomenų mainų formatas.

NPM – Node Package Manager. JavaScript programinės įrangos paketų tvarkytuvė, naudojama diegiant ir valdant bibliotekas „Node.js“ projektuose.

REST – Representational State Transfer. Architektūrinis API kūrimo principas, grindžiamas HTTP protokolu.

SMTP – Simple Mail Transfer Protocol. Protokolas, naudojamas elektroninio pašto siuntimui tarp serverių.

VRAM – Video Random Access Memory. Vaizdo plokštės darbinė atmintis.

Terminai:

Mašininis mokymasis – metodų visuma, leidžianti kompiuteriui mokytis iš duomenų ir priimti sprendimus be tiesioginio programavimo.

Transformeris – Neuroninių tinklų architektūra.

Encoder-decoder architektūra – modelių architektūra, kurioje vienas tinklas koduoja įvestį, o kitas generuoja išvestį.

Duomenų praturtinimas – metodai, skirti padidinti duomenų kiekį.

Generatyvus modelis – modelis, gebantis generuoti naują turinį remiantis išmokta informacija.

Mažai išteklių turinti kalba – kalba, kuriai trūksta kokybiškų NLP išteklių, tokių kaip anotuoti duomenys, modeliai ar įrankiai.

Modelių priderinimas (angl. *fine-tuning*) – Iš anksto apmokyto modelio parametrų priderinimas prie konkrečios užduoties arba kalbos.

Back-translation – duomenų praturtinimo metodas, kai tekstas išverčiamas į kitą kalbą ir atgal, siekiant sukurti naujas teksto formuluotes.

Parafrazavimas – duomenų praturtinimo metodas, kai tekstas sutrumpinamas, išlaikant originalią prasmę.

Sinonimų keitimas – duomenų praturtinimo metodas, kuriame kai kurie žodžiai tekste pakeičiami jų sinonimais.

Įvadas

Natūralios kalbos apdorojimo (angl. *Natural Language Processing, NLP*) srityje mašininio mokymosi modelių taikymas tapo svarbiu aspektu, leidžiančiu automatizuoti užduotis, kurias anksčiau atlikdavo žmonės. Viena iš tokių užduočių – klausimų kūrimas ir atsakymas į juos pagal pateiktą kontekstą. Ši užduotis yra ypač svarbi, siekiant kurti pažangias mokymosi sistemas, automatizuotas konsultavimo, techninės pagalbos platformas.

Pastaraisiais metais reikšmingai išaugo didelių kalbų modelių naudojimas, leidžiantis generuoti aukštos kokybės tekstus įvairiomis kalbomis. Tačiau nepaisant spartaus šios srities vystymosi, mažųjų kalbų, tokių kaip lietuvių, apdorojimas vis dar susiduria su daugybe iššūkių. Riboti duomenų kiekiai ir nepakankamas modelių priderinimo lietuvių kalbai lemia, kad klausimų kūrimo ir atsakymo užduotys atliekamos mažesniu tikslumu nei anglų ar kitomis didesnėmis kalbomis.

Dėl šios priežasties kyla poreikis ieškoti sprendimų, kurie leistų pagerinti lietuvių kalbos apdorojimo kokybę esant ribotiems resursams – be milžiniškų duomenų rinkinių ar itin didelių kompiuterinių resursų. Viena galimybė tai padaryti – naudoti duomenų praturtinimą (angl. *data augmentation*), kuris leidžia išplėsti turimus duomenis ir taip pagerinti modelių gebėjimą apdoroti tekstą.

Darbo aktualumas

Lietuvių kalba, kaip mažiau resursų turinti kalba, NLP srityje susiduria su dideliais iššūkiais: trūksta viešai prieinamų kokybiškų duomenų rinkinių, o esami didieji modeliai nėra optimaliai jai pritaikyti. Dėl to dažnai gaunami tekstai būna sintaksiškai ar semantiškai netikslūs. Kadangi lietuvių kalba turi sudėtingą morfologinę struktūrą, metodai, naudojami anglų ar kitomis kalbomis, dažnai neveikia pakankamai gerai.

Atsižvelgiant į šias problemas, aktualu ištirti, kaip taikant duomenų praturtinimo metodus galima pagerinti klausimų kūrimo ir atsakymo sistemų kokybę lietuvių kalba, remiantis jau egzistuojančiais modeliais ir naudojant ribotus duomenų išteklius.

Problema

Nors daugelis kalbų modelių palaiko lietuvių kalbą, praktikoje jie dažnai nepasižymi aukšta kokybe. Klausimų kūrimo ir atsakymo kokybė yra ribota dėl duomenų trūkumo ir modelių nepakankamo priderinimo šiai kalbai. Todėl reikia rasti būdus, kaip pagerinti modelių veikimą papildomais metodais.

Tikslas ir uždaviniai

Darbo tikslas – ištirti mašininio mokymosi modelių taikymą klausimų kūrimui ir atsakymui lietuvių kalba, taikant duomenų praturtinimo metodus siekiant pagerinti rezultatų kokybę.

Uždaviniai:
1. Atlikti klausimų kūrimo ir atsakymo užduočių analizę, įvertinant dabartines technologijas ir jų taikymą mažai resursų turinčioms kalboms.
2. Išanalizuoti esamus duomenų praturtinimo metodus ir įvertinti jų pritaikomumą lietuvių kalbos NLP užduotims.

3. Sukurti eksperimentinę aplinką, leidžiančią taikyti pasirinktus metodus klausimų kūrimo ir atsakymo modelių tobulinimui.
4. Atlikti eksperimentus su baziniu ir praturtintais duomenų rinkiniais, įvertinti modelių veikimo skirtumus.
5. Suformuluoti išvadas apie duomenų praturtinimo metodų efektyvumą mažai resursų turinčiose kalbose.

Darbo naujumas

Šiame darbe siūlomas praktinis metodas, kaip pagerinti klausimų kūrimo ir atsakymo kokybę lietuvių kalba, pasitelkiant duomenų praturtinimą. Skirtingai nei dauguma tyrimų, orientuotų į anglų kalbą ar naudojančių itin didelius duomenų bei skaičiavimo resursus, šiame darbe siekiama parodyti, kaip su ribotais resursais galima pasiekti reikšmingų kokybės patobulinimų, taikant duomenų manipuliavimo būdus.

Darbo struktūra

Pirmajame skyriuje pateikiama analitinė dalis, kurioje apžvelgiamos esamos klausimų kūrimo ir atsakymo technologijos, iššūkiai mažai resursų turinčioms kalboms bei duomenų praturtinimo metodai.

Antrajame skyriuje aprašomas sukurtas programinis sprendimas – internetinė sistema, leidžianti naudotojams naudoti mašininio mokymosi modelius lietuvių kalba.

Trečiajame skyriuje pateikiama tyrimo eiga ir planas.

Ketvirtajame skyriuje pateikiami ir interpretuojami tyrimo rezultatai, aptariami iššūkiai bei suformuojamos išvados.

1. Analitinė dalis

Analitinėje dalyje aptariamos pagrindinės klausimų kūrimo ir atsakymo generavimo technologijos, ypač mažai išteklių turinčiomis kalbomis.

1.1. Klausimų kūrimo ir atsakymo technologijų apžvalga

Mašininio mokymosi metodų taikymas klausimų kūrimo (angl. *Question Generation, QG*) ir atsakymo į klausimus (angl. *Question Answering, QA*) užduotims pastarąjį dešimtmetį tapo viena iš aktyviausiai tyrinėjamų sričių natūralios kalbos apdorojime. Šios technologijos leidžia ne tik automatizuoti mokymosi ir informacijos paieškos procesus, bet ir sudaro pagrindą pažangesnių dialogo sistemų vystymui [1,2,3].

Ankstyvieji darbai šioje srityje daugiausia rėmėsi simboliniais metodais (angl. *symbolic/rule-based methods*): iš anksto apibrėžtomis gramatikos taisyklėmis ir semantiniais šablonais. Tokie sprendimai dažniausiai buvo taikomi labai siauruose kontekstuose ir buvo riboti savo pritaikomumu. Išpopuliarėjus gilaus mokymosi modeliams (angl. *Deep Learning*), situacija pasikeitė. Pasirodžius transformatorių architektūrai (angl. *transformer*), modeliai kaip T5 [4] ir BART [5] parodė puikų gebėjimą generuoti klausimus ir atsakyti į juos remiantis pateiktu tekstu [6].

Modelis T5-3B (Text-to-Text Transfer Transformer) buvo ypač reikšmingas, nes suvienodino įvairias NLP užduotis paversdamas jas į „tekstas į tekstą" užduotis [4]. Tai leido taikyti vienodą architektūrą tiek klausimų kūrimui, tiek atsakymų paieškai. Panašiai, BART modelis, naudojantis „encoder-decoder" architektūrą, tapo populiarus tekstų generavimo užduotyse.

Daugiakalbiai modeliai – mT5 (101 kalba) [7], mBART-25 ir mBART-50 išplėtė šias galimybes vidutinėmis ir mažomis kalbomis. Praktikoje pastebima, kad šie modeliai be papildomo priderinimo (angl. *fine-tuning*) dažnai generuoja gramatiškai neteisingus tekstus mažai išteklių turinčiomis kalbomis [8,9].

Lietuvių kalbai skirtų modelių pasirinkimas ilgą laiką buvo ribotas. 2020 m. atsirado Baltijos kalboms priderintas modelis LitLat-BERT (Base 110 M param.) [10], 2024 metais – LT-Llama-2-13B-Instruct – pastarasis leidžia generuoti aukštesnės kokybės lietuvišką tekstą, bet nėra lengvai naudojamas mažai resursų turinčiose sistemose [11]. Taip pat pastaruoju metu pasirodė lietuviški Q&A duomenų rinkiniai kaip Lithuanian QA V1 (13848 klausimų ir atsakymų porų) [11] bei QALD-9-plus (654 klausimų ir atsakymų porų) [12]. Nepaisant šių teigiamų pokyčių, turimų duomenų apimtis vis dar išlieka per maža efektyviam generatyvių modelių treniravimui, todėl duomenų praturtinimas tampa būtinu žingsniu siekiant pagerinti lietuvių kalbos QG/QA sistemų kokybę [13,14].

Taigi, nors pasaulyje klausimų kūrimo ir atsakymo technologijos yra pakankamai pažengusios, jų taikymas lietuvių kalbai vis dar reikalauja papildomų pastangų, siekiant išnaudoti turimus modelius ir infrastruktūrą efektyviai.

1.2. Mažai išteklių turinčių kalbų problematika

Natūralios kalbos apdorojimo srityje didžiausi proveržiai pasiekti anglų kalba, turinčia itin didelius teksto kiekius ir tyrimams skirtus resursus. Tačiau mažiau paplitusios kalbos, tokios kaip lietuvių, susiduria su iššūkiais, kurie tiesiogiai lemia mašininio mokymosi modelių veikimo kokybę.

Mažai resursų turinčios kalbos (angl. *low-resource languages*) apibrėžiamos kaip tokios, kurioms trūksta didelių viešai prieinamų duomenų rinkinių, specializuotų modelių ar infrastruktūros, leidžiančios efektyviai vykdyti NLP užduotis [8,15]. Šiose kalbose resursų trūkumas riboja galimybes treniruoti pažangius neuroninius tinklus.

Lietuvių kalba yra mažai resursų turinčios kalbos pavyzdys. Nors egzistuoja tam tikri resursai, kaip CLARIN-LT duomenų rinkiniai [16], šių duomenų nepakanka siekiant optimaliai apmokyti generatyvius modelius tokioms užduotims kaip klausimų kūrimas ar atsakymas į klausimus. Be to, lietuvių kalba pasižymi dideliu morfologiniu sudėtingumu – linksniavimu, asmenavimu, gausia priesagų sistema bei lanksčia sakinio struktūra, dėl ko modeliams tampa sudėtingiau generuoti taisyklingą ir natūraliai skambantį tekstą.

Dar vienas aspektas, apsunkinantis darbą su mažai resursų turinčiomis kalbomis, yra modelių išankstinis treniravimas (angl. *pretraining*). Dauguma dabartinių didžiųjų kalbų modelių, net jei teoriškai palaiko šimtus kalbų, praktikoje didžiąją dalį treniravimo resursų paskiria daugiausiai naudojamoms kalboms, tokioms kaip anglų, ispanų ar kinų. Tai lemia, kad mažesnėms kalboms tenka labai ribotas duomenų kiekis ir juos būna sunkiau priderinti šioms kalboms.

Atsižvelgiant į tai, siekiant pagerinti lietuvių kalbos klausimų kūrimo ir atsakymo kokybę, būtina imtis papildomų veiksmų: taikyti specialius modelių treniravimo metodus, pasitelkti duomenų praturtinimą, ar naudoti rankiniu būdu sukurtus ar dirbtinai padidintus duomenų rinkinius.

1.3. Esami modeliai ir jų taikymas mažai resursų turinčioms kalboms

Mašininio mokymosi modeliai, skirti klausimų kūrimo ir atsakymo užduotims, sparčiai išsivystė, ypač po transformerių architektūros atsiradimo. Tačiau net ir modernūs modeliai, tokie kaip T5, mT5 ar BART, yra orientuoti į daug resursų turinčias kalbas. Dėl šios priežasties jų veikimas mažai išteklių turinčioms kalboms, tarp jų ir lietuvių kalbai, dažnai būna ribotas.

T5 modelis, treniruotas anglų kalba, parodė pažangius rezultatus įvairiose NLP užduotyse [4]. Jo daugiakalbė versija, mT5, buvo ištreniruota naudojant 101 kalbą naudojant „Common Crawl“ pagrindu sudarytą „C4 corpus“ rinkinio atitikmenį. Tačiau treniravimo ir modelio optimizavimo metu mažiau resursų turinčiomis kalbomis buvo skirta mažiau dėmesio, todėl, nors mT5 palaiko lietuvių kalbą, jo generuojami klausimai dažnai būna nerišlūs.

Panaši situacija pastebima ir su mBART modeliu – jis buvo treniruojamas naudojant 25 kalbų duomenimis, tarp jų – ir lietuvių. Modelis orientuotais į vertimo užduotį ir papildomai priderintas tekstų generavimo užduočiai, todėl teksto generavimas naudojant šį modelį nėra tikslus, ypač mažai resursų turinčiomis kalbomis [17].

Pastaruoju metu atsirado modeliai, skirti Baltijos šalių kalboms, pavyzdžiui, LitLat-BERT. Tai modelis, skirtas žemo lygmens užduotims – klasifikacijai, vardų atpažinimui, žodžių įterpimui. Dėl architektūros apribojimų jis nėra tiesiogiai pritaikytas klausimų kūrimui ar atsakymui [10].

Dar viena reikšminga iniciatyva yra LT-Llama-2 modelis. Tai lietuvių kalbai adaptuotas atvirojo kodo Llama-2 modelis, skirtas generatyvioms NLP užduotims, tarp jų ir klausimų kūrimui. Nors jo dydis ir galingumas leidžia pasiekti geresnius rezultatus, praktinis jo naudojimas reikalauja didelių

skaičiavimo resursų, kurie ne visuomet yra prieinami. LT-Llama-2 yra 13 B parametrų modelis[11,18], kuriam reikalingi dideli skaičiavimo resursai.

Taip pat egzistuoja ir keli lietuvių kalbai priderinti teksto generavimo modeliai. Vienas aktualus modelis – t5-base-lithuanian-news-summaries-17, kuris yra apmokytas teksto santraukų užduočiai i, naudojant daugiau nei 2 milijonus straipsnių [19]. Šio modelio paskirtis yra architektūriškai artima klausimų kūrimui, todėl modelis galėtų būti tinkamas šiai užduočiai su papildomu pritaikymu.

Kitas modelis – ByT5-Lithuanian-gec-100h, kuris skirtas klaidų taisymo užduočiai [20]. Šis modelis pasižymi baitiniu tokenizavimu, kuris padeda efektyviai identifikuoti klaidas žodžiuose. Modelis treniruotas klaidų taisymui lietuvių kalba – todėl jo perkėlimas į klausimų kūrimo ar atsakymo generavimo užduotis nėra tiesioginis, bet galimas su papildomu priderinimu.

Pagrindinių modelių palyginimas pateikiamas 1 lentelėje.

1 lentelė. Modelių palyginimas

Modelis	**Pritaikytos kalbos**	**Tikslas**	**Tinkamumas QG/QA lietuvių kalba**	**Parametrų sk.**
T5	Anglų	Universalus teksto apdorojimas	Ribotas, reikalauja priderinimo kitomis kalbomis	60M – 11B
mT5	101 kalba	Daugelio kalbų teksto apdorojimas	Galimas, bet sunkiai pritaikomas sudėtingoms užduotims	300M – 13B
mBART	25 kalbos (mBART-50 – 50 kalbų)	Teksto generavimas, vertimas	Galimas, bet ribotas lietuvių kalbai teksto generavime	680M
LitLat-BERT	Lietuvių, latvių	Klasifikacija, NER, užpildymas	Netinkamas tiesioginiam QG/QA	270M
LT-Llama-2	Lietuvių	Teksto generavimas	Tinkamas, bet reikalauja daug resursų	13B
t5-base-lithuanian-news-summaries-17	Lietuvių	Teksto santraukų generavimas	Priderintas lietuvių kalbai, reikalauja papildomo priderinimo klausimų generavimo ir atsakymo užduočiai.	220M
ByT5-Lithuanian-gec-100h	Lietuvių	Teksto klaidų taisymas	Priderintas lietuvių kalbai, reikalauja papildomo priderinimo klausimų generavimo ir atsakymo užduočiai.	580M

Nors egzistuoja keletas modelių, kuriuos galima pritaikyti lietuvių kalbai, jų efektyvus taikymas klausimų kūrimo ir atsakymo užduotims dažnai reikalauja papildomo duomenų praturtinimo ir priderinimo metodų, siekiant kompensuoti ribotą pradinių lietuviškų duomenų kiekį.

1.4. Modelių priderinimas

Daugelis šiuolaikinių NLP modelių, tokių kaip T5 ar mT5, yra iš anksto apmokyti (angl. *pretrained*) dideliais bendro pobūdžio duomenų rinkiniais. Nors toks apmokymas leidžia modeliams įgyti bendrą

gebėjimą generuoti tekstą [21], specifinėms užduotims ar konkrečioms kalboms reikalingas papildomas priderinimas [22].

Modelio priderinimas – tai modelio parametrų adaptavimas konkrečiai užduočiai ar duomenų rinkiniui. Šiame etape modelis mokosi pagal konkretesnius duomenis ir gali išmokti tam tikros kalbos ar srities bruožus. Lyginant su pirminiu mokymu, modelio priderinimas reikalauja mažiau duomenų ir skaičiavimo išteklių, tačiau gali reikšmingai pagerinti modelio veikimą konkrečiame kontekste.

Modelių priderinimas ypač svarbus mažai išteklių turinčiomis kalbomis, nes išankstinio apmokymo metu šioms kalboms dažnai tenka mažesnė treniravimo duomenų dalis. Tinkamai parengtas duomenų rinkinys bei papildomas priderinimas leidžia pagerinti tokių modelių teksto generavimo kokybę ir sukurti modelius, priderintus tam tikrai užduočiai [23].

1.5. Duomenų rinkinių sudarymas

Modelių priderinimo kokybė tiesiogiai priklauso nuo naudojamų duomenų rinkinių sudėties [24]. Kuriant duomenų rinkinius QA ir QG užduotimis dažniausiai pateikiamas kontekstas, klausimas ir atsakymas, kuris gali būti randamas kontekste. Ši struktūra leidžia duomenų rinkinį naudoti ir klausimų generavimo, ir klausimų atsakymo užduotimis.

Vienas iš dažniausiai naudojamų klausimų ir atsakymų rinkinių – SQuAD (Stanford Question Answering Dataset) – suformuotas remiantis „Wikipedia“ tekstais ir žmonių anotacijomis [25]. Šis rinkinys pasižymi aukšta kokybe, bet tokių rinkinių sudarymas reikalauja daug žmogiškųjų resursų. Dėl šios priežasties daug dėmesio skiriama automatizuotiems rinkinių kūrimo metodams [22,26], kurie leidžia generuoti duomenis naudojant galingesnius kalbos modelius, arba taisyklėmis grįstą logiką.

Kuriant duomenų rinkinius mažai išteklių turinčiomis kalbomis, dažnai pasitelkiami kitos kalbos rinkiniai (pvz., SQuAD) ir taikomi automatinio vertimo įrankiai [27,28,29,30]. Tokia praktika, nors ir leidžia greitai sukurti lokalizuotus rinkinius, kelia iššūkių dėl semantinio tikslumo ir vertimų kokybės. Vertimo metu gali būti prarasta sintaksinė ar semantinė struktūra, o atsakymai gali tapti nesuderinti su nauju kontekstu.

Nustatyta, kad generatyvūs modeliai gali būti naudojami ne tik klausimų kūrimui, bet ir viso duomenų rinkinio generavimui [31,32]. Tokie metodai ypač naudingi mažai išteklių turinčioms kalboms, kur trūksta resursų, reikalingų duomenų rinkinius kurti rankiniu būdu. Šiuolaikiniai didieji kalbų modeliai (pvz., ChatGPT, Gemini) taip pat gali būti naudojami papildomam duomenų generavimui [33]. Generuojant papildomus duomenis galima smarkiai padidinti duomenų kiekį ir įvairumą. Tačiau šis metodas taip pat reikalauja didelės kokybės kontrolės, siekiant išvengti semantinių klaidų ar prasmės praradimo [22,34,35].

Taip pat naudojami hibridiniai metodai – kai duomenys generuojami automatiškai, bet vėliau tikrinami arba koreguojami žmogaus [30]. Tai leidžia užtikrinti aukštesnę kokybę, tačiau išlaikyti didesnį automatizavimo lygį.

Galima teigti, kad efektyvus duomenų rinkinių sudarymas yra vienas iš svarbiausių NLP modelių treniravimo etapų. Ypač mažai išteklių kalboms svarbu taikyti metodus, kurie leidžia automatizuoti duomenų rinkinio sudarymo procesą.

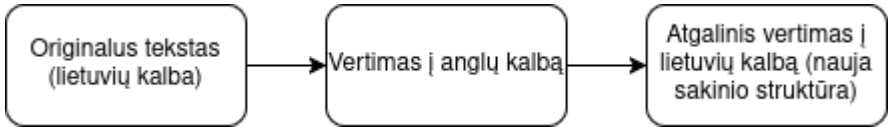
1.6. Duomenų praturtinimo metodai

Mažai išteklių turinčiose kalbose dažnai susiduriama su problema, kad modelių treniravimui nepakanka kokybiškų duomenų. Vienas iš būdų šiai problemai spręsti yra duomenų praturtinimas, kuris leidžia padidinti turimą duomenų kiekį, išlaikant semantinę prasmę ir gramatinį taisyklingumą [32].

NLP srityje taikomi įvairūs duomenų praturtinimo metodai.

Atvirkštinis vertimas

Atvirkštinis vertimas (angl. *back-translation*) – plačiai naudojamas metodas, taikomas tiek mašininio vertimo, tiek generavimo užduotyse [36]. Šio metodo esmė – pradinį tekstą versti į kitą kalbą ir tuomet atgal į originalią kalbą. Tokiu būdu sukuriamas naujas tekstas, išsaugantis semantinę prasmę, bet dažnai turintis kitokią sakinio struktūrą [22]. Atvirkštinio vertimo metodas iliustruojamas 1 pav.



1 pav. Atvirkštinio vertimo metodo schema

Parafrazavimas

Parafrazavimas (angl. *paraphrasing*) reiškia teksto perrašymą kitais žodžiais, išsaugant tą pačią prasmę [37]. Tai leidžia padidinti duomenų įvairovę, o modeliai išmoksta geriau suprasti skirtingas sakinio formuluotes [38]. Gali būti treniruojami specializuoti parafrazavimo modeliai šiam duomenų praturtinimo būdui.

Sinonimų keitimas

Paprastesnis duomenų praturtinimo būdas – sinonimų keitimas (angl. *synonym replacement*). Šiuo atveju atskiri žodžiai tekste pakeičiami jų sinonimais, siekiant išlaikyti sakinio prasmę. Nors metodas paprastas, taikant jį būtina atsižvelgti į galimas semantines klaidas, ypač morfologiškai sudėtingose kalbose [38].

Duomenų praturtinimas tampa esminiu įrankiu siekiant pagerinti mažai resursų turinčių kalbų modelių gebėjimą generuoti taisyklingus klausimus ir atsakymus. Kiekvienas iš aprašytų metodų turi savo stipriąsias ir silpnąsias puses, todėl dažniausiai geriausių rezultatų pasiekiama juos tarpusavyje derinant.

1.7. Generuoto teksto kokybės vertinimo metrikos

Vertinant klausimų kūrimo ir atsakymų generavimo užduotis, būtina naudoti objektyvias metrikas, leidžiančias kiekybiškai įvertinti sugeneruoto teksto atitikimą tiksliniam rezultatui. Šiuo tikslu dažniausiai taikomos BLEU (angl. *Bilingual Evaluation Understudy*)[39] ir ROUGE (angl. *Recall-Oriented Understudy for Gisting Evaluation*)[40] metrikos. Jos leidžia automatizuotai įvertinti generavimo modelių kokybę remiantis žodžių ar jų sekų panašumu į tikslinį tekstą [41,42].

BLEU metrika

BLEU yra viena pirmųjų ir plačiausiai naudojamų automatinių vertinimo metrikų, originaliai sukurta vertimo užduotims. Ši metrika remiasi sugeneruoto ir tikslinio teksto n-gramų sutapimų analize. N-grama yra *n* gretimų simbolių seka tam tikra tvarka. Kuo daugiau sugeneruoto sakinio n-gramų sutampa su tikslinėmis, tuo didesnė BLEU reikšmė.

Bendroji BLEU formulė išreiškiama taip [39]:

$$BLEU = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \cdot log\, p_n\right) \ ;$$

čia:

- p_n – n-gramų tikslumas,
- w_n – svoriai,
- N – didžiausias naudojamas n-gramų ilgis
- BP – baudos koeficientas už sakinio ilgį:

$$BP = \begin{cases} 1, & jei\ c > r \\ e^{(1-r/c)}, & jei\ c \leq r \end{cases};$$

čia c yra sugeneruoto sakinio ilgis, o r – atskaitos sakinio ilgis

BLEU metrika efektyviai vertina sintaksinį atitikimą, tačiau yra kritikuojama už tai, kad nevertina semantikos – du sakiniai gali turėti tą pačią prasmę, bet gauti skirtingą BLEU įvertį dėl skirtingos žodžių tvarkos ar sinonimų naudojimo. Vertinant semantiką, BLEU gali būti nepakankamas, todėl taip pat taikomos ROUGE metrikos.

ROUGE metrikos

ROUGE metrikų rinkinys dažniausiai naudojamas tekstų apibendrinimo užduotyse, tačiau plačiai taikomas ir generavimo NLP užduotyse. Dažnai naudojamos šios ROUGE versijos:

- ROUGE-1 – žodžių (unigramų) atitikimas;
- ROUGE-2 – bigramų (dviejų žodžių sekų) atitikimas;
- ROUGE-L – ilgiausios bendros sekos (LCS, angl. *Longest Common Subsequence*) ilgis.

ROUGE-N formulė apibrėžiama taip [40]:

$$ROUGE\text{-}N = \frac{\sum_{ref \in Refe} \sum_{gram_n \in ref} Count_{match}(gram_n)}{\sum_{ref \in Ref} \sum_{gram_n \in ref} Count(gram_n)} \ ;$$

čia:

- *Ref* – tikslinis tekstas,
- $Count_{match}(gram_n)$ – kiek kartų n-grama pasikartoja ir sugeneruotame, ir tiksliniame tekste,
- $Count(gram_n)$ – bendras n-gramų kiekis tiksliniame tekste.

ROUGE-L remiasi ilgiausios bendros sekos (LCS) analize. Jos formulė:

$$ROUGE\text{-}L = \frac{LCS(X,Y)}{len(X)} \ ;$$

čia:

- *LCS(X, Y)* – ilgiausios bendros sekos ilgis tarp tikslinio ir sugeneruoto teksto,
- *len(X)* – tikslinio teksto ilgis

1.8. Susiję darbai ir tyrimai

Klausimų kūrimo ir atsakymo užduotys mažai išteklių turinčiose kalbose ilgą laiką buvo mažai tyrinėtos. Dauguma didžiųjų tyrimų projektų orientavosi į anglų kalbą, o bandymai pritaikyti analogiškas sistemas kitoms kalboms buvo vykdomi tik pastaraisiais metais [8,43].

Vienas iš aktualiausių projektų – TyDi QA rinkinys [8], skirtas klausimų atsakymui įvairiomis mažesnėmis kalbomis, tarp jų ir suomių, bengalų, suahilių. Tyrimai parodė, kad net pažangūs daugiakalbiai modeliai (pvz., mT5) be specialaus priderinimo šiose kalbose generavo prastesnius atsakymus nei anglų kalboje. Tam pačiam rinkiniui geresnių rezultatų buvo pasiekta taikant duomenų praturtinimą.

Kitas svarbus projektas – XQuAD (Cross-lingual Question Answering Dataset), kuriame buvo išverstos anglų kalbos klausimų-atsakymų poros į kitas kalbas, tarp jų ir ispanų, vokiečių, arabų [43]. Šis rinkinys įrodė, kad vertimo pagrindu galima sėkmingai perkelti klausimų-atsakymų duomenų rinkinius tarp kalbų, tačiau vertimas neišvengiamai sukelia tam tikrą semantikos praradimą.

Mažai išteklių turinčiomis kalbomis pritaikytų projektų vis dar mažai. Tyrimas, atliktas teksto generavimui suomių kalba, parodė, kad net naudojant verstus kitų kalbų duomenų rinkinius, galima ženkliai pagerinti modelių kokybę tiek klausimų kūrimo, tiek atsakymo užduotyse [44].

Lietuvių kalbos kontekste buvo sukurti keli svarbūs resursai, tokie kaip QALD-9-plus išplėtimas [12], kuriuose siekiama plėtoti klausimų-atsakymų sistemų pritaikymą lietuvių kalbai. Vis dėlto šių rinkinių dydis ir duomenų įvairovė vis dar riboti, todėl būtina taikyti papildomus praturtinimo metodus.

Pagrindinių susijusių darbų apžvalga pateikiama 2 lentelėje.

2 lentelė. Pagrindiniai klausimų kūrimo ir atsakymo tyrimai mažai išteklių turinčioms kalboms

Tyrimas / rinkinys	**Kalba (-os)**	**Metodai**	**Pastebėjimai**
TyDi QA	11 kalbų	Modeliai, duomenų praturtinimas	Daugiakalbiai modeliai be papildomo priderinimo prastai veikia mažose kalbose
XQuAD	10 kalbų	Vertimas	Vertimas padeda, bet galima semantinio tikslumo praradimas
Ilmari Kylliäinen, Roman Yangarber tyrimas	Suomių	Vertimas	Versti duomenų rinkiniai reikalauja papildomos peržiūros.
QALD-9-plus	Lietuvių (išplėstas)	Semantinių klausimų sistema	Pradiniai rezultatai teigiami, bet apimtis nedidelė

2. Projektinė dalis

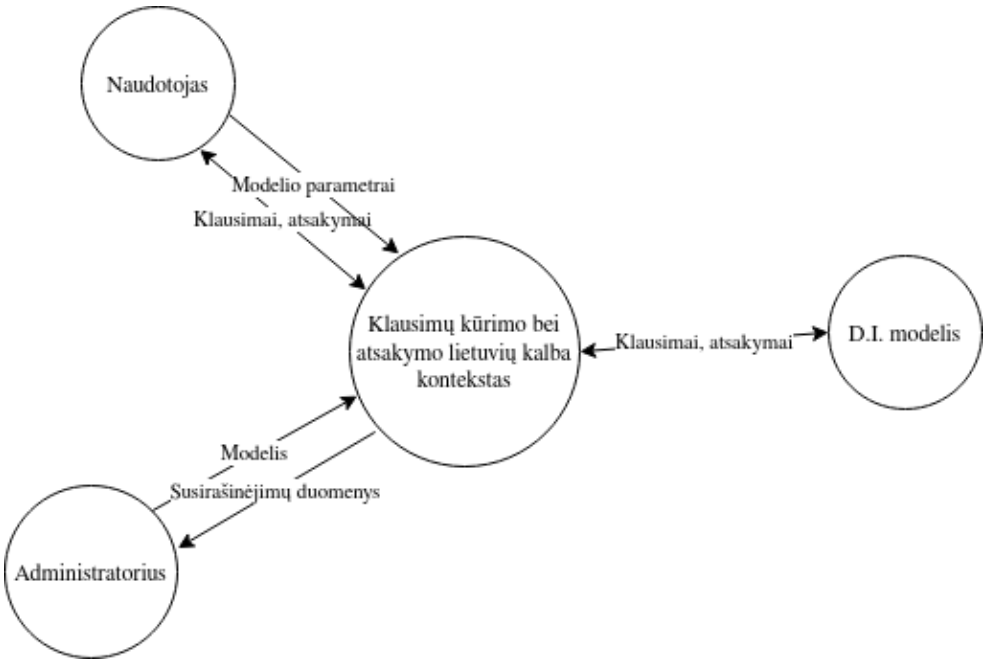
Projektinėje dalyje aprašoma sukurta sistema, leidžianti naudotojams naudoti dirbtinio intelekto modelius, sukurtais klausimų kūrimo ir atsakymų generavimo užduotims lietuvių kalba. Sistema veikia kaip žiniatinklio platforma, turinti galimybę generuoti klausimus pagal pateiktą tekstą, atsakyti į klausimus remiantis kontekstu bei leisti naudotojui treniruoti individualų modelį, priderintą specifiniams poreikiams.

Šioje dalyje pateikiamas sistemos kontekstas, architektūriniai sprendimai, komponentų sąveika, duomenų valdymo ir diegimo mechanizmai.

2.1. Sistemos veiklos kontekstas

Sukurta sistema skirta lietuvių kalba veikiančių klausimų kūrimo ir atsakymo D.I. modelių sąveikai su naudotojais, leidžianti jiems generuoti klausimus, gauti atsakymus ir atlikti modelių treniravimą bei vertinimą. Sistema apima du pagrindinius naudotojų tipus – įprastus naudotojus ir administratorius – bei sąveiką su D.I. modeliais, kurie naudojami teksto generavimo užduotims.

Sistemos veiklos konteksto schema pavaizduota 2 pav. Joje išskiriami keturi pagrindiniai komponentai: naudotojas, administratorius, D.I. modelis ir sistema. Naudotojas sąveikauja su sistema pateikdamas tekstus, klausimus ir modelio parametrus, o atgal gauna sugeneruotus atsakymus. Administratorius turi išplėstines galimybes – jis gali prižiūrėti duomenų srautus, atnaujinti modelius ir vykdyti sistemos modelių atnaujinimą. D.I. modeliai atlieka klausimų ir atsakymų generavimą pagal naudotojo inicijuotas užklausas.



2 pav. Sistemos veiklos konteksto diagrama

Sistemos veikla yra orientuota į du pagrindinius veiklos srautus: naudotojo sąveiką su testo generavimo modeliais ir administratoriaus vykdomą modelių priežiūrą bei duomenų apdorojimą. Šie srautai yra esminiai užtikrinant sistemos funkcionalumą ir tobulinimą. Sistemos veiklos komponentų sąveika detaliai aprašyta 3 lentelėje.

3 lentelė. Sistemos veiklos konteksto aprašymas

Srautas	**Kryptis**	**Srauto aprašymas**
Modelio parametrai	Iš naudotojo	Naudotojai kuria priderintus D.I. modelius pateikiant modelio parametrus bei susijusius duomenis.
Klausimai, atsakymai	Iš naudotojo	Naudotojai pateikia tekstą, kuriam nori gauti atsakymą.
Klausimai, atsakymai	Į naudotoją	Sistemos sugeneruoti atsakymai į pateiktas užklausas pateikiami naudotojui.
Modelis	Iš administratoriaus	Administratorius atnaujina pagrindinį sistemos modelį.
Susirašinėjimų duomenys	Į administratorių	Administratorius tvarko naudotojų pateiktas užklausas ir D.I. modelio atsakymus šiuos duomenis naudojant D.I. modelio įvertinimui bei tobulinimui.
Klausimai, atsakymai	Į D.I. modelį	Pateikiama įvestis D.I. modeliui.
Klausimai, atsakymai	Iš D.I. modelio	Pateikiama D.I. modelio sugeneruotas tekstas.

2.2. Naudotojai ir panaudojimo atvejų modelis

Sistema skirta sąveikai su klausimų kūrimo ir atsakymo modeliais lietuvių kalba, todėl numato keletą naudotojų tipų, kurių kiekvienas turi skirtingą prieigos lygį ir funkcionalumo ribas. Funkcionalumas priklauso nuo to, ar naudotojas yra prisijungęs, turi mokamą paskyrą, ar yra administratorius.

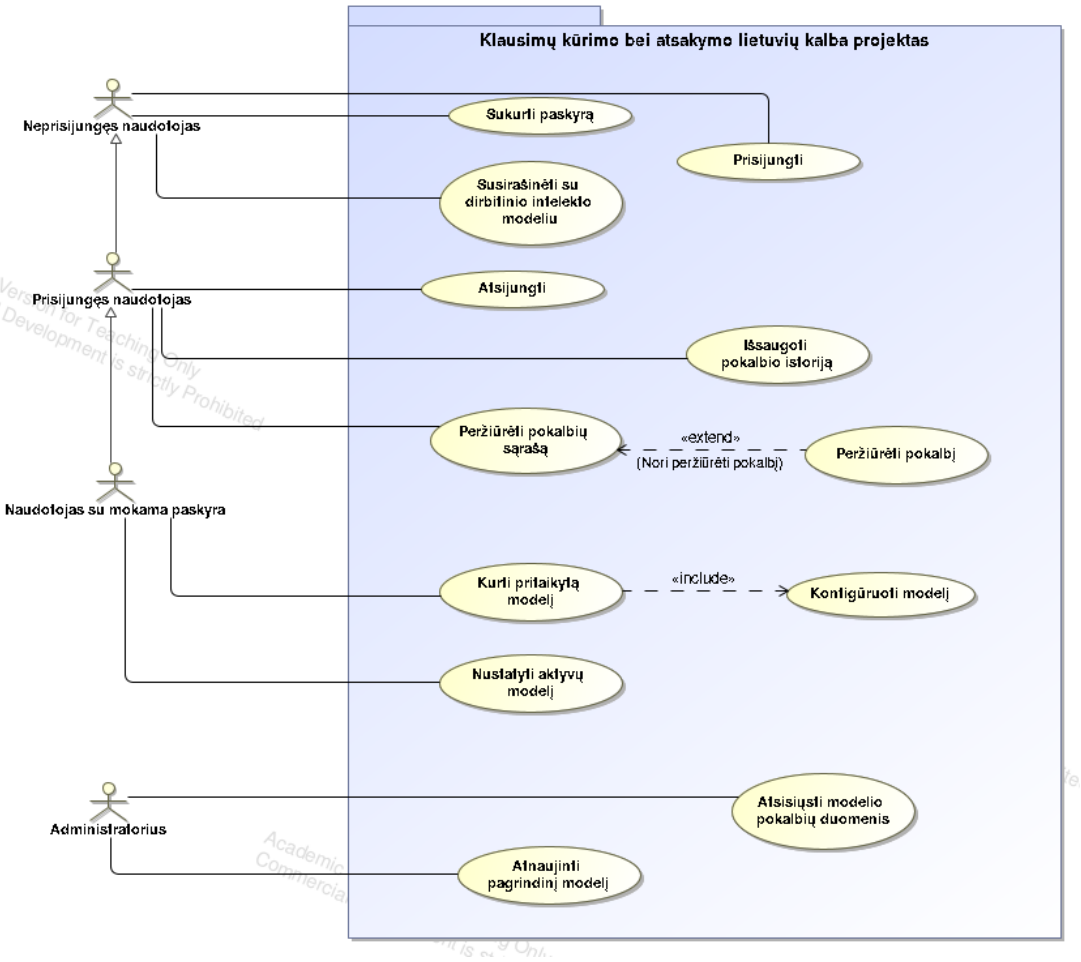
Neprisijungęs naudotojas gali:
- PA1 „Susikurti paskyrą“;
- PA2 „Prisijungti“;
- PA3 „Susirašinėti su dirbtinio intelekto modeliu“.

Prisijungęs naudotojas gali:
- PA4 „Atsijungti“;
- PA5 „Išsaugoti pokalbio istoriją“;
- PA6 „Peržiūrėti pokalbių sąrašą“;
- PA7 „Peržiūrėti pokalbį“.

Naudotojas su mokama paskyra gali:
- PA8 „Kurti pritaikytus modelius“;
- PA9 „Konfigūruoti modelius“;
- PA10 „Nustatyti aktyvų modelį, kuris bus naudojamas pokalbiams“.

Administratorius turi didžiausią prieigą ir gali:
- PA11 „Atsisiųsti modelių pokalbių duomenis analizei ar vertinimui“;
- PA12 „Atnaujinti pagrindinį modelį“.

Šių naudotojų veiksmus sisteminiu požiūriu atspindi panaudojimo atvejų diagrama, pateikiama 3 pav. Ji vizualiai parodo visų naudotojų sąveikas su sistema, veiksmų priklausomybes ir galimus scenarijų išplėtimus.



3 pav. Panaudojimo atvejų diagrama

2.3. Funkciniai ir nefunkciniai reikalavimai

2.3.1. Papildomi funkciniai reikalavimai

Pagrindiniai sistemos funkciniai reikalavimai specifikuojami panaudojimo atvejų sąraše, jiems taip pat sukurta detali specifikacija. Taip pat keliami papildomi funkciniai reikalavimai (žr. 4 lentelė, 5 lentelė, 6 lentelė, 7 lentelė).

4 lentelė. Modelio tobulinimo galimybių reikalavimas

Reikalavimas #:	13	**Reikalavimo tipas:**	10	**Įvykis/panaudojimo atvejis #:**	11, 12
Aprašymas:		Turi būti galimybė atnaujinti ir tobulinti pagrindinį sistemos teksto generavimo modelį.			
Pagrindimas:		Sistemą paviešinus naudotojų pokalbiai su dirbtinio intelekto modeliu gali būti naudojami kaip duomenys toliau tobulinti modelį.			
Šaltinis:		Sistemos projektuotojas			
Tinkamumo kriterijus:		Galima parsisiųsti sistemoje saugomus duomenis bei atnaujinti sistemos naudojamą modelį.			
Užsakovo tenkinimas:		3		Užsakovo netenkinimas:	2

Prioritetas:	Vidutinis		
Priklausomybės:	-	Konfliktai:	-
Papildoma medžiaga:	-		
Istorija:	Užregistruotas 2023 Kovo 1 d.		

5 lentelė. Atsakymo pergeneravimo reikalavimas

Reikalavimas #:	14	**Reikalavimo tipas:**	10	**Įvykis/panaudojimo atvejis #:**	3
Aprašymas:	Turi būti galimybė pergeneruoti D.I. modelio sugeneruotą tekstą.				
Pagrindimas:	Pirmasis pateiktas D.I. modelio atsakymas gali netenkinti naudotojo.				
Šaltinis:	Sistemos projektuotojas				
Tinkamumo kriterijus:	Pateiktą atsakymą galima atmesti, sugeneravus naują atsakymą jis turėtų nesutapti su ankstesniu atsakymu.				
Užsakovo tenkinimas:	3			Užsakovo netenkinimas:	3
Prioritetas:	Vidutinis				
Priklausomybės:	3			Konfliktai:	-
Papildoma medžiaga:	-				
Istorija:	Užregistruotas 2023 Kovo 1 d.				

6 lentelė. Duomenų rinkimo atsisakymo reikalavimas

Reikalavimas #:	15	**Reikalavimo tipas:**	10	**Įvykis/panaudojimo atvejis #:**	3
Aprašymas:	Turi būti galimybė atsisakyti duomenų saugojimo susirašant su D.I. modeliu.				
Pagrindimas:	Naudotojas, saugodamas savo privatumą, gali nenorėti, kad jo pokalbiai su D.I. modeliu būtų matomi kitų asmenų.				
Šaltinis:	Sistemos projektuotojas				
Tinkamumo kriterijus:	Naudotojo, atsisakiusio duomenų saugojimo, žinutės nėra pasiekiamos perkrovus puslapį.				
Užsakovo tenkinimas:	2			Užsakovo netenkinimas:	4
Prioritetas:	Aukštas				
Priklausomybės:	3			Konfliktai:	-
Papildoma medžiaga:	-				
Istorija:	Užregistruotas 2023 Kovo 1 d.				

7 lentelė. Programų sąsajos reikalavimas

Reikalavimas #:	16	**Reikalavimo tipas:**	10	**Įvykis/panaudojimo atvejis #:**	3, 5, 6, 7, 10
Aprašymas:	Turi būti galimybė atlikti veiksmus, susijusius su D.I. sąveika naudojant programinę sąsają.				

Pagrindimas:	Naudotojai, norintys integruoti sistemą į savo programas tai turi galėti atlikti naudojant programinę sąsają.		
Šaltinis:	Sistemos projektuotojas		
Tinkamumo kriterijus:	Panaudojimo atvejus galima atlikti naudojant programinę sąsają, nenaudojant naudotojo sąsajos.		
Užsakovo tenkinimas:	2	Užsakovo netenkinimas:	2
Prioritetas:	Vidutinis		
Priklausomybės:	-	Konfliktai:	-
Papildoma medžiaga:	-		
Istorija:	Užregistruotas 2023 Kovo 1 d.		

2.3.2. Nefunkciniai reikalavimai

Sistema taip pat turi atitikti tam tikrus nefunkcinius reikalavimus. Jie apima naudotojo patogumą, sistemos atsakymo laiką, elgseną esant didelei apkrovai, duomenų tvarkymo saugumą ir prieigos ribojimus. Pagrindiniai nefunkciniai reikalavimai pateikti 8 lentelėje.

8 lentelė. Sistemos nefunkciniai reikalavimai

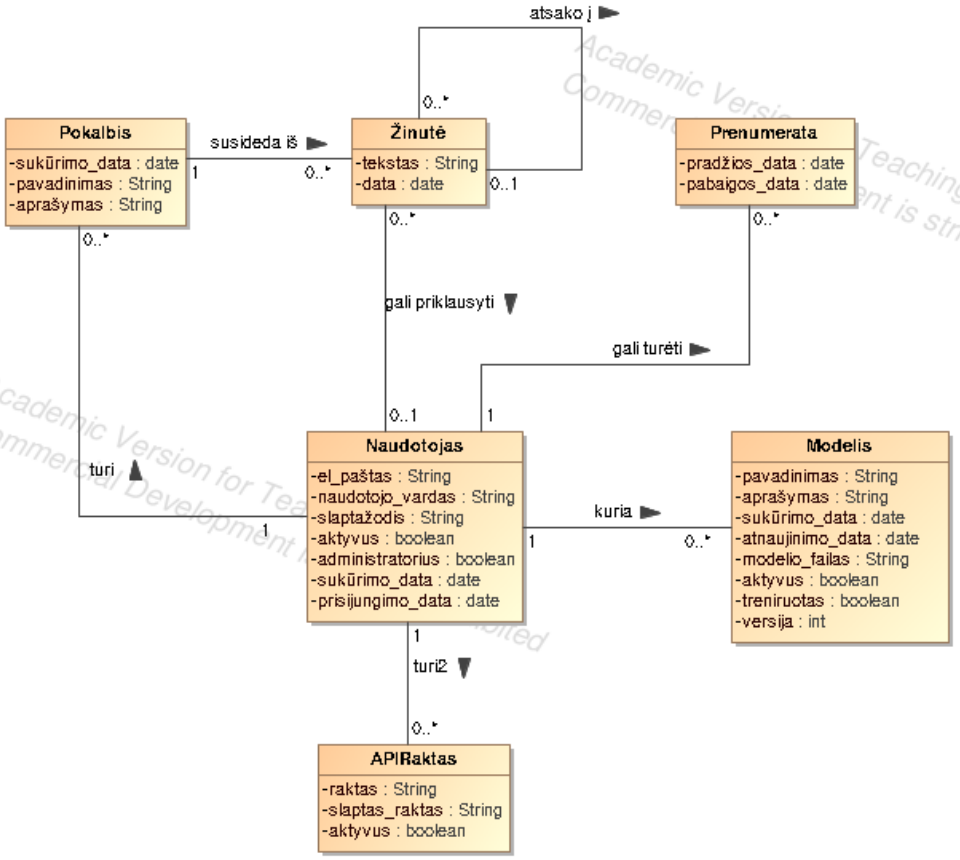
Reikalavimas	**Kategorija**
Sistemos dizainas turi būti minimalistinis	Išvaizda
Sistema turi būti nesudėtinga naudoti	Panaudojamumas
Prisijungusiam naudotojui sistema turi prisijungimą prisiminti tame pačiame įrenginyje	Panaudojamumas
Atsakymas turi būti pradėtas generuoti ne vėliau kaip per 20 sek. nuo užklausos pateikimo	Vykdymo charakteristikos
Sistema turi bandyti palaikyti pokalbį net esant nerišliam tekstui	Vykdymo charakteristikos
Turi būti galima apdoroti bent 10 užklausų vienu metu	Vykdymo charakteristikos
Sistemos pagrindinis D.I. modelis turi būti tobulinamas	Priežiūra
Sistema neturi reikalauti naudotojo palaikymo	Priežiūra
Naudotojas gali aktyvuoti tik savo sukurtus modelius	Saugumas
Tik administratoriai gali peržiūrėti visų naudotojų pokalbius	Saugumas
Sistemoje saugomi duomenys turi atitikti GDPR reikalavimus.	Teisė

Šie reikalavimai buvo įvertinti projektavimo metu ir turėjo įtakos tiek architektūriniams sprendimams, tiek naudotojo sąsajos ir paslaugų elgsenos modeliavimui.

2.4. Duomenų vaizdas

Duomenų vaizdas leidžia suprasti, kaip informacija kaupiama, kokie esminiai objektai yra sistemoje ir kokie ryšiai tarp jų egzistuoja.

Sistemos duomenų struktūra pavaizduota esybių-ryšių diagramoje (žr. 4 pav.), kuri atskleidžia pagrindinius duomenų tipus bei jų tarpusavio ryšius.



4 pav. Sistemos esybių-ryšių diagrama

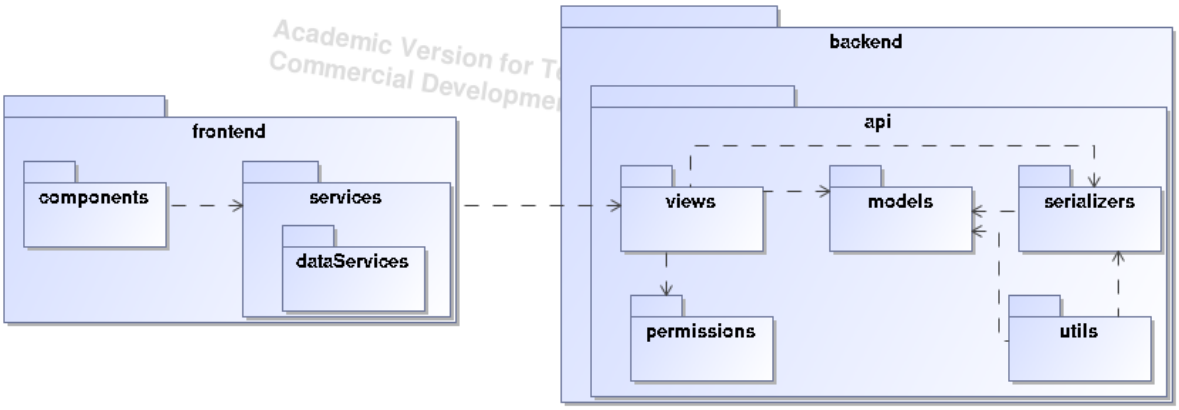
Diagramoje pateikiamos šios esybės:

- Naudotojas – saugo informaciją apie naudotoją, jo būseną ir turimą prieigą.
- Pokalbis – apima naudotojo pokalbius. Vienas naudotojas gali turėti daug pokalbių.
- Žinutė – Vienas pokalbis susideda iš kelių žinučių, o žinutė gali priklausyti konkrečiam naudotojui. Jei žinutė nepriklauso naudotojui, reiškia, kad ji buvo sugeneruota D.I. Modelio.
- Modelis – nurodo naudotojo sukurtus modelius.
- API raktas – leidžia naudotojams pasiekti sistemą per API. Saugo raktą, slaptą reikšmę ir aktyvumo būseną.
- Prenumerata – susieta su naudotoju, naudojama prieigos teisėms valdyti.

2.5. Sistemos statinis vaizdas

2.5.1. Paketų apžvalga

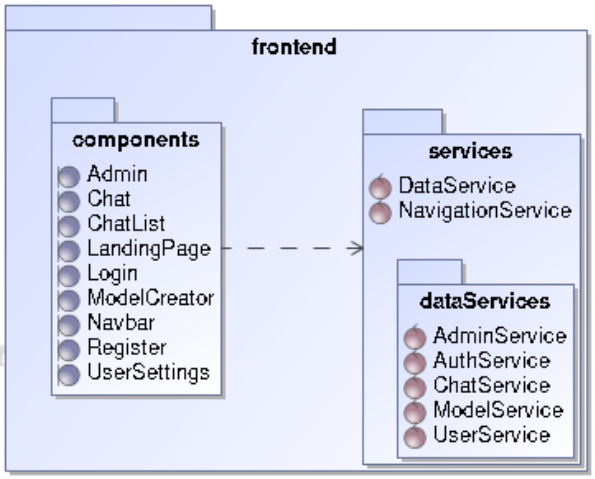
Projektas yra grupuojamas į du pagrindinius paketus (žr. 5 pav.).



5 pav. Sistemos abstraktus išskaidymas į paketus

Klientinė dalis skaidoma į du atskirus paketus (žr. 6 pav.):

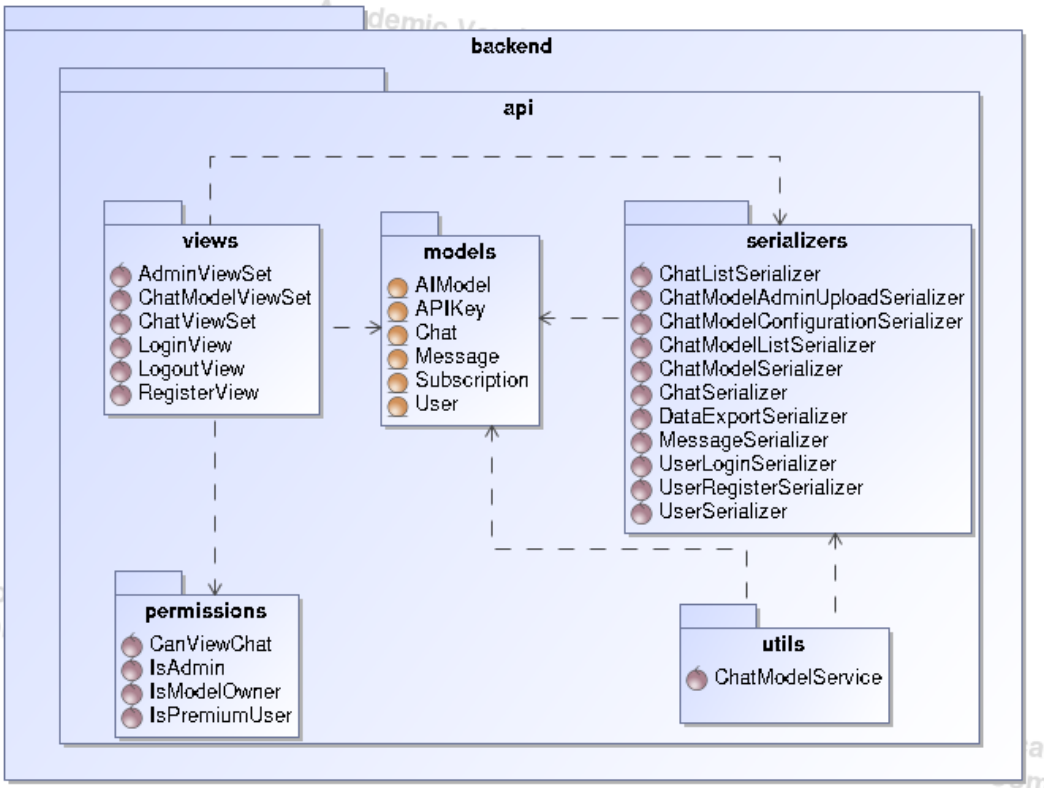
- *components* – naudotojo sąsają apibrėžiantys komponentai;
- *services* – apibrėžia duomenų keitimosi su serverine sistemos dalimi, duomenų apdorojimo, autorizacijos logiką. Vidiniame *dataServices* pakete talpinamos klasės, atsakingos už tam tikros duomenų grupės apdorojimą.



6 pav. Sistemos klientinės dalies paketai

Serverinė dalis turi pagrindinį *api* paketą, kuris toliau skaidomas į šiuos vidinius paketus (žr. 7 pav.):

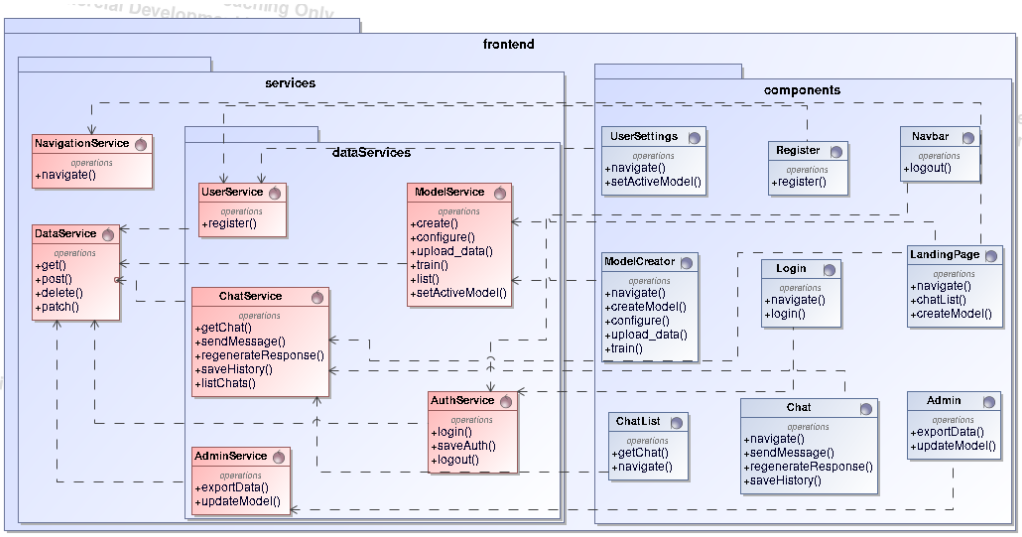
- *models* – duomenų bazės modelių deklaracija objektiniu pavidalu;
- *views* – programavimo sąsajos apibrėžimas;
- *permissions* – prieigos teisių konfigūracija;
- *serializers* – klasės, atliekančios duomenų atvaizdavimą JSON formatu, duomenų validaciją bei objektų kūrimą;
- *utils* – kontrolerių klasės, realizuojančios įvairią sistemos logiką.



7 pav. Sistemos serverinės dalies paketai

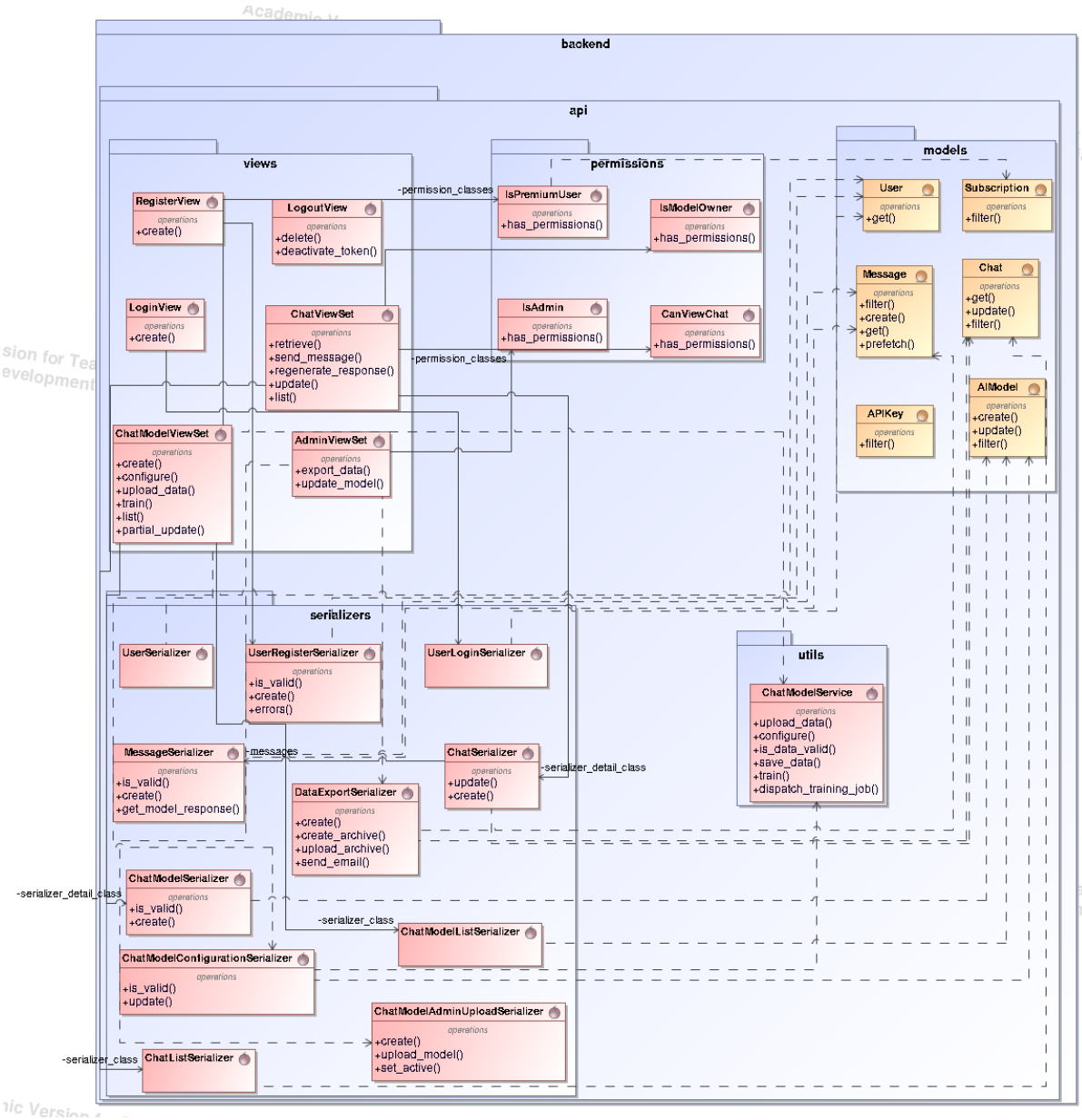
2.5.2. Paketų detalizavimas

Detalizuota klientinės dalies klasių diagrama pateikiama 8 pav. Ribinės klasės naudoja atitinkamų resursų kontrolerių klases iš *dataServices* paketo. Klasės šiame pakete naudoja *DataService* klasę, kuri atlieka HTTP užklausas į serverinę sistemos dalį. Gavus atsaką iš *DataService* klasės, atitinkamos *dataServices* klasės atlieka rezultatų interpretavimą bei duomenų struktūravimą, tai leidžia ribinėmis klasėmis atvaizduoti rezultatus. Taip pat *NavigationService* klasė atlieka navigacijos veiksmus, kurie leidžia naudotojui pasiekti skirtingas sistemos funkcijas.



8 pav. Sistemos klientinės dalies klasės

Detalizuota serverinės dalies klasių diagrama pateikiama 9 pav. Pakete *views* esančios kontrolerių klasės apibrėžia sistemos programinės sąsajos galutinius taškus skirtingiems sistemos resursams. Kiekviena iš šių klasių gali turėti *permission_classes* atributą, pagal kurį nustatoma naudotojo prieigos teisių apribojimas. Naudotojas gali būti autorizuojamas pagal API raktų porą, arba prisijungimo duomenis. Šis funkcionalumas yra įgyvendintas naudojamo karkaso ribose. Kiekviena iš *views* paketo klasių taip pat susiejama su vienu ar daugiau klasių iš *serializers* paketo, kurios gali paversti duomenis iš modelio klasės objekto į JSON formatą ir atvirkščiai. Jos atsakingos už pagrindinę vykdomą logiką su duomenimis. Šio paketo klasės naudoja atitinkamas modelių klases bei kitas klases iš *serializers* paketo. Pakete *utils* išskiriamos kontrolerių klasės, siekiant išskaidyti sudėtingus veiksmus.



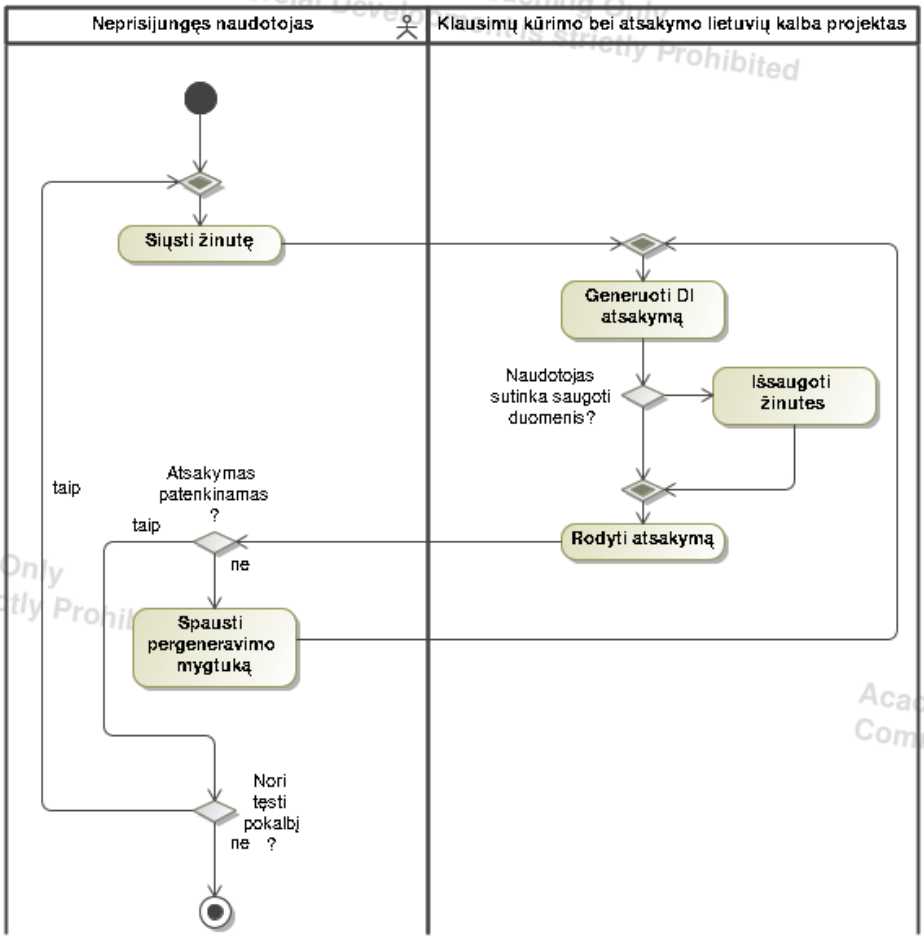
9 pav. Sistemos serverinės dalies klasės

2.6. Sistemos dinaminis vaizdas

Sistemos dinaminis elgesys aprašomas veiklos ir sekų diagramomis, kurios iliustruoja pagrindinių panaudojimo atvejų eigą bei komponentų sąveiką laike. Šiame skyriuje pateikiamos veiklos ir sekų diagramos svarbiausiems sistemos scenarijams: susirašinėjimą su dirbtinio intelekto modeliu, priderinto modelio kūrimą ir konfigūravimą.

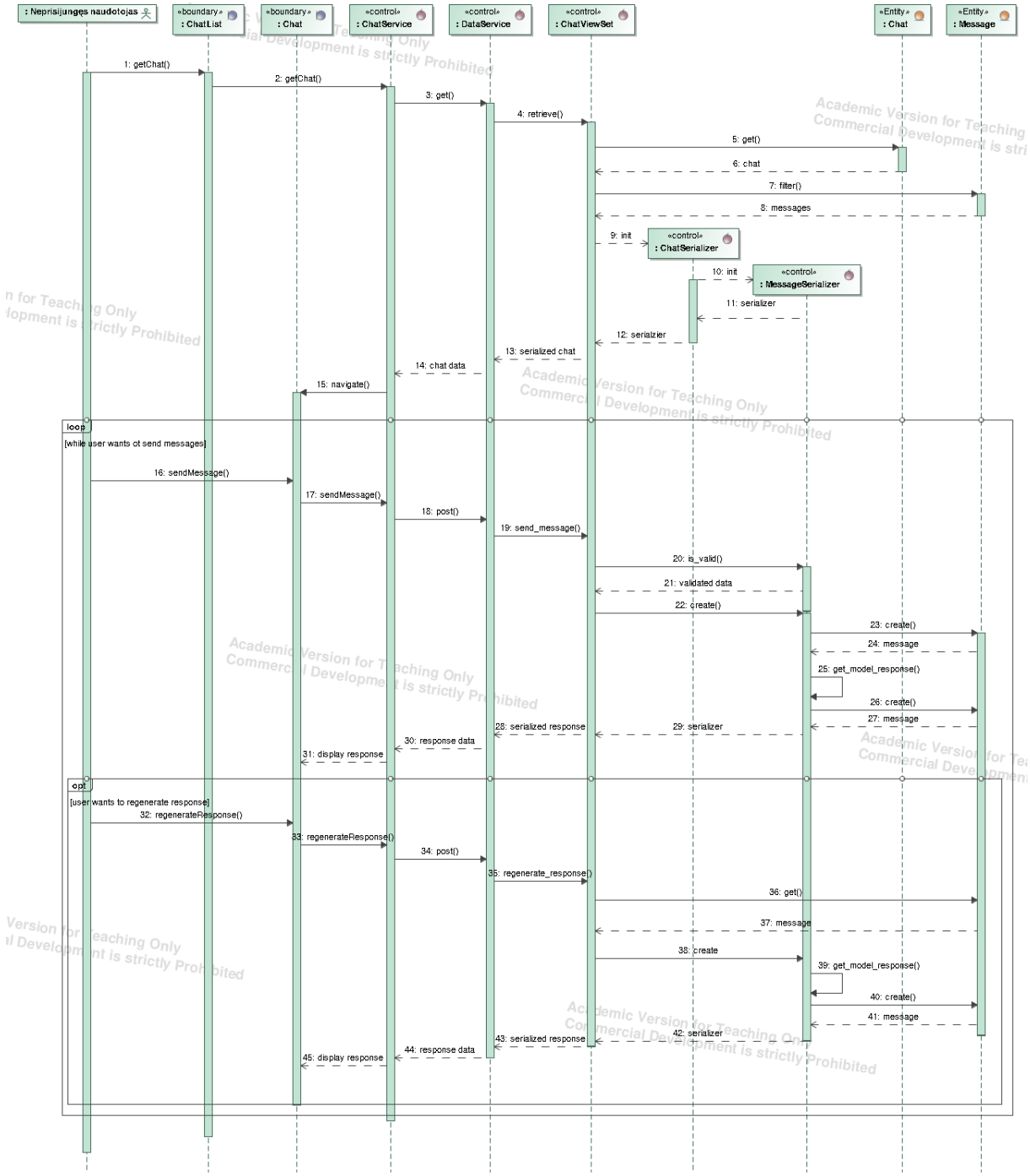
2.6.1. Susirašinėjimas su D.I. modeliu

Šis atvejis apibūdina pagrindinį naudotojo veiksmą – pateikti tekstą ir gauti atsakymą iš dirbtinio intelekto modelio. Veiklos eiga aprašoma veiklos diagramoje (žr. 10 pav.), kurioje pavaizduoti žingsniai nuo žinutės siuntimo iki atsakymo generavimo ir istorijos išsaugojimo.



10 pav. Veiklos diagrama – „Susirašinėti su D.I. modeliu“

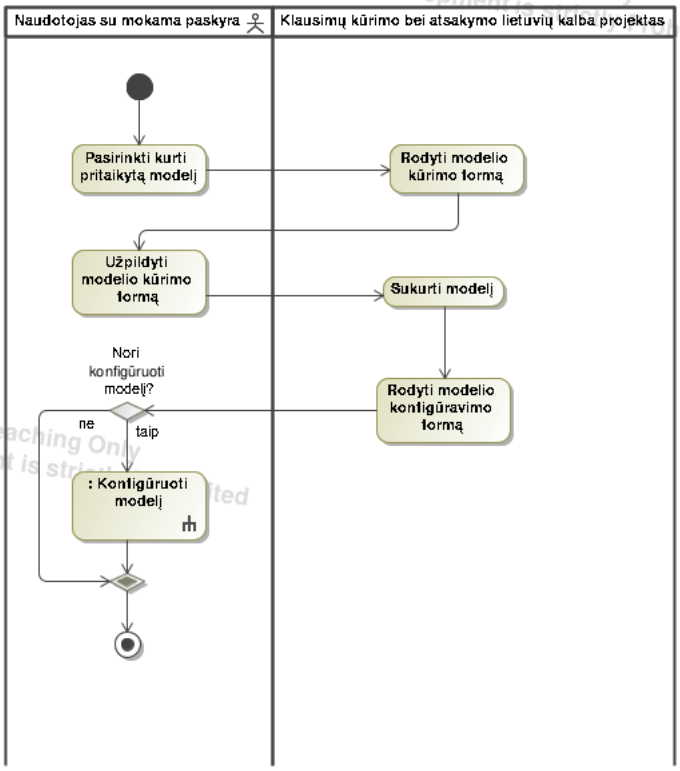
Sekų diagrama (žr. 11 pav.) detalizuoja šio proceso techninį įgyvendinimą: naudotojo sąveiką su naudotojo sąsajos komponentais ir serverine dalimi.



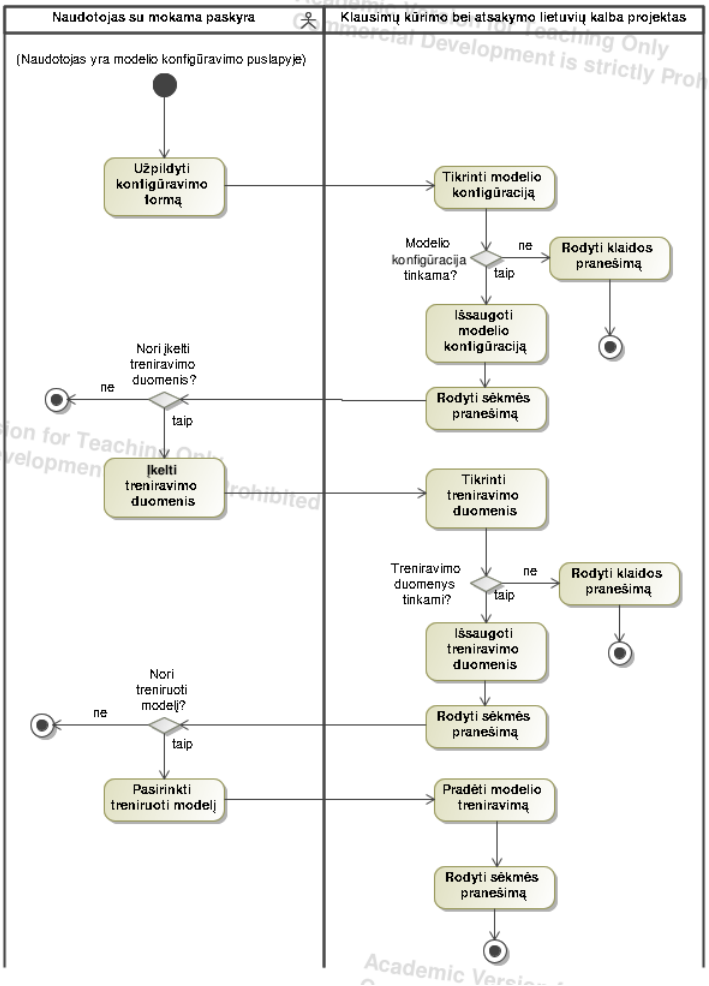
11 pav. Sekų diagrama – „Susirašinėti su D.I. modeliu“

2.6.2. Priderinto modelio kūrimas ir konfigūravimas

Šis panaudojimo atvejis leidžia naudotojui, turinčiam mokamą paskyrą, susikurti, konfigūruoti ir treniruoti savo modelį. Procesas prasideda modelio kūrimo inicijavimu ir aprašytas veiklos diagramoje (žr. 12 pav.), kurioje taip pat pavaizduota galimybė konfigūruoti modelį kaip papildomą veiksmą. Modelio konfigūravimo procesas taip pat apibūdinamas veiklos diagramoje (žr. 13 pav.).

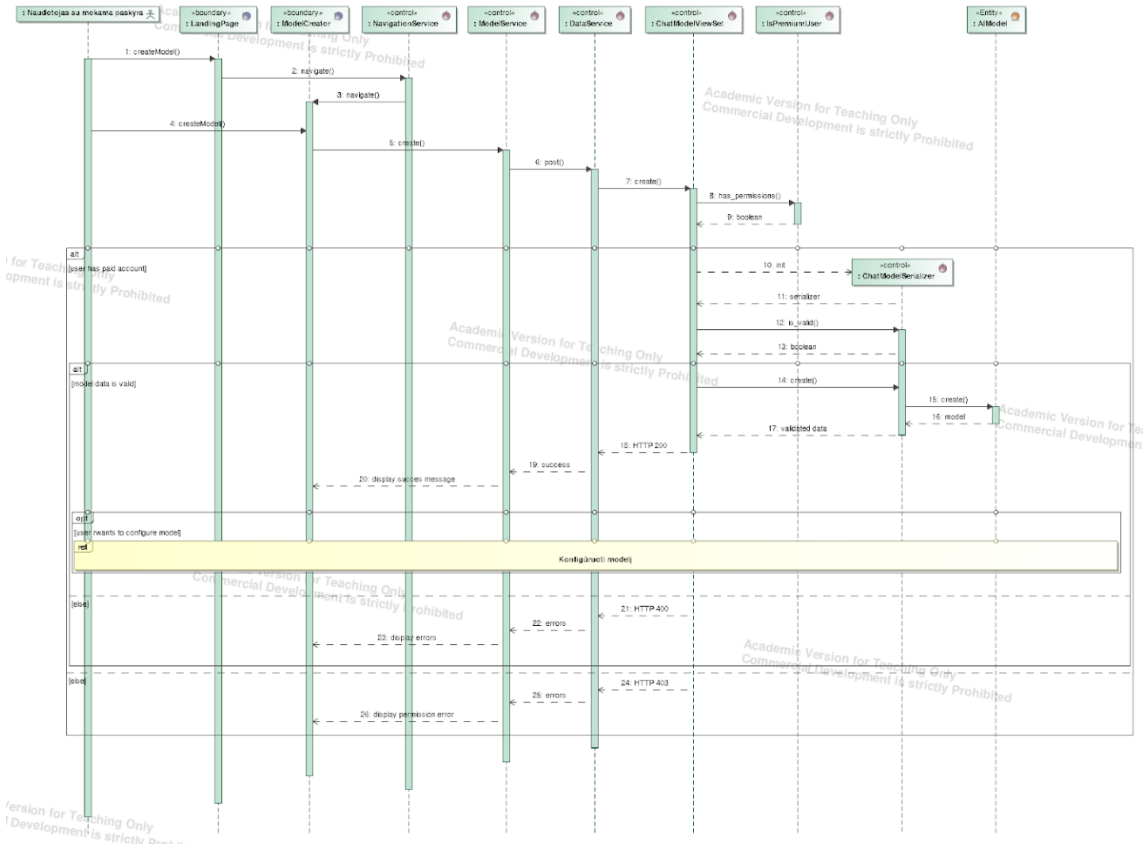


12 pav. Veiklos diagrama – „Kurti pritaikytą modelį“

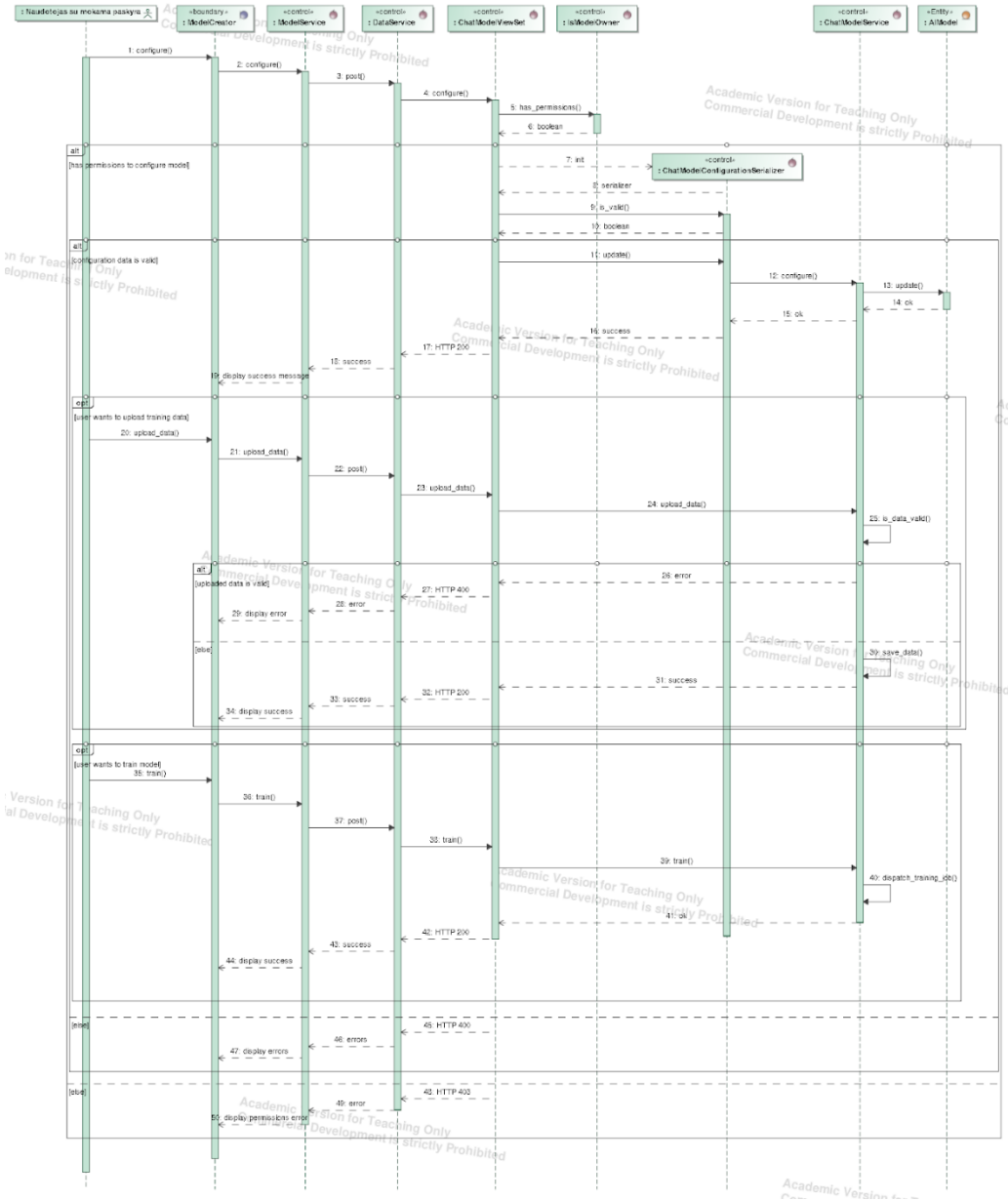


13 pav. Veiklos diagrama – „Konfigūruoti modelį“

Sekų diagramose apibūdinama tiksli komponentų sąveika šių veiksmų metu. Modelio kūrimas apima formos pateikimą ir modelio sukūrimą (žr. 14 pav.), o modelio konfigūracija – išplėstinę konfigūraciją, duomenų rinkinio įkėlimą ir treniravimo inicijavimą (žr. 15 pav.)



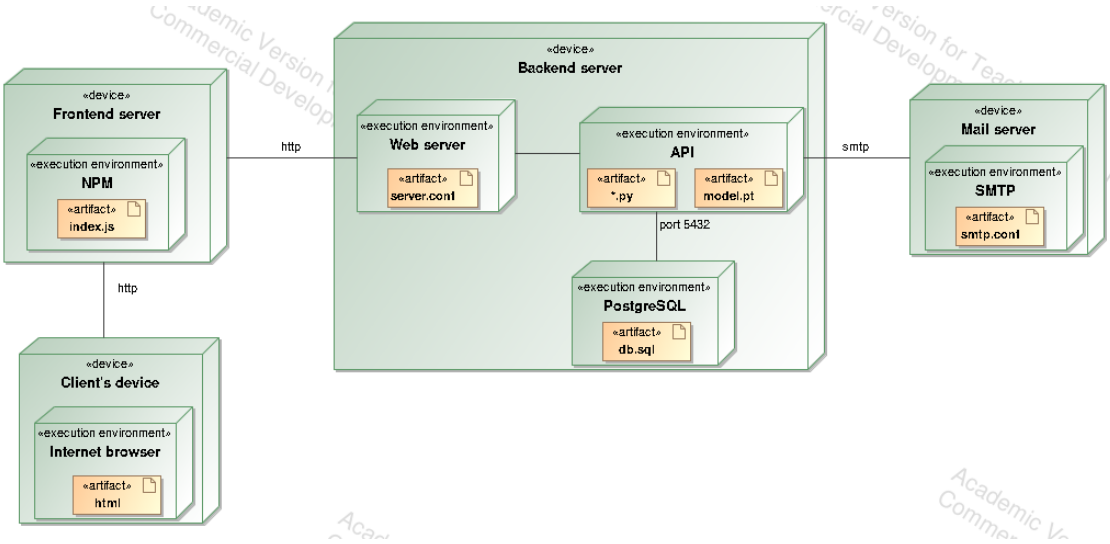
14 pav. Sekų diagrama – „Kurti modelį“



15 pav. Sekų diagrama – „Konfigūruoti modelį“

2.7. Diegimo architektūra

Diegimo architektūra vaizduoja, kaip skirtingi sistemos komponentai yra išdėstyti fizinėje ir loginėje infrastruktūroje. Sistema susideda iš kelių nepriklausomų komponentų, kurių kiekvienas turi aiškiai apibrėžtą funkciją ir veikia skirtinguose įrenginiuose ar konteineriuose (žr. 16 pav.).



16 pav. Sistemos diegimo diagrama

Naudotojas sąveikauja su sistema naudodamas naršyklę, kurioje pasiekiama naudotojo sąsaja, pateikiama *frontend* serverio. Šis serveris paleidžiamas naudojant *NPM* aplinką. Užklausos siunčiamos į *backend* serverį, kuriame veikia keli komponentai:

- Tinklo serveris, atsakingas už užklausų nukreipimą į API sluoksnį.
- API komponentas, kuriame veikia Python programinis kodas
- Duomenų bazė, kurioje saugomi duomenys.

Sistema taip pat integruota su atskiru elektroninio pašto serveriu. Visi komponentai veikia tarpusavyje nepriklausomai, o komunikacija tarp jų vykdoma HTTP ir SMTP protokolais. Tokia architektūra leidžia sistemą lengvai plėsti ir išlaikyti aiškų komponentų paskirstymą tarp serverių.

3. Tyrimo dalis

3.1. Įvadas

Šiame tyrime siekiama įvertinti duomenų praturtinimo metodų poveikį klausimų kūrimo ir atsakymų generavimo modelių kokybei lietuvių kalboje, pasitelkiant egzistuojančius transformerių architektūros modelius. Atsižvelgiant į tai, kad lietuvių kalba turi mažai duomenų, viena iš pagrindinių kliūčių efektyviam tokių modelių priderinimui yra nepakankami aukštos kokybės duomenys. Todėl tyrimo objektu pasirinktas jų treniravimui naudojamų duomenų sudarymo bei praturtinimo būdų galimybės.

Tyrimo tikslas – išanalizuoti, kaip skiriasi klausimų kūrimo ir atsakymo generavimo modelių veikimas priklausomai nuo naudoto duomenų rinkinio sudarymo būdo (originalus ar praturtintas). Tyrimas remiasi hipoteze, kad duomenų praturtinimas gali ženkliai pagerinti modelių veikimo rezultatus net esant mažam duomenų kiekiui.

Tyrimo uždaviniai:
1. Sudaryti originalų lietuvių kalbos klausimų–atsakymų duomenų rinkinį, remiantis automatizuotu tekstų analizės ir generavimo procesu.
2. Praturtinti šį rinkinį taikant augmentacijos metodus.
3. Priderinti (angl. *fine-tune*) transformerių modelius naudojant originalų ir praturtintą duomenų rinkinius.
4. Išmatuoti modelių veikimo kokybę.
5. Palyginti gautus rezultatus su nepriderinto modelio atsakymais.
6. Įvertinti, ar duomenų praturtinimas reikšmingai pagerina rezultatų kokybę.

3.2. Duomenų rinkinių sudarymas

Tyrimui sukurtas lietuvių kalbos klausimų-atsakymų duomenų rinkinys, sudarytas dviem etapais. Pirmasis etapas apėmė duomenų rinkimą ir klausimų generavimą iš lietuviškų Vikipedijos straipsnių, o antrasis – anglų kalbos klausimų rinkinio (Stanford Question Answering Dataset, SQuAD) pritaikymą lietuvių kalbai naudojant mašininį vertimą.

Abu duomenų rinkiniai sukurti JSON formatu, kur kiekvienas įrašas turi kontekstą, klausimą bei atsakymą.

3.2.1. Vikipedijos duomenų rinkinys

Pirmasis rinkinys buvo sudarytas automatizuotai. Buvo išrinkti straipsniai iš pasirinktų kategorijų: Istorija, Geografija, Mokslas, Kultūra, Menas, Kalbos. Iš kiekvienos kategorijos parinkta po 10 straipsnių. Surinkti straipsnių tekstai buvo apdoroti šalinant nereikalingas struktūrines teksto dalis („Šaltiniai“, „Išnašos“, „Nuorodos“ ir kt.), žymes, citatas ir nereikalingus simbolius. Tam naudotos reguliariosios išraiškos (žr. 9 lentelė.).

9 lentelė. Naudotos teksto valymo taisyklės

Taisyklės aprašymas	**Reguliarioji išraiška / veiksmas**
Pašalinami tekstai skliausteliuose	re.sub(r"\[.*?\]", "", text)
Pašalinami HTML žymėjimai	re.sub(r"<.*?>", "", text)

Pašalinami keli nauji paragrafai	re.sub(r"\n\s*\n+", "\n", text)
Apkarpomi tušti tarpai eilutėse	line.strip() kiekvienai eilutei
Normalizuojamas tarpas	re.sub(r"\s+", " ", text)
Pašalinami nelietuviški simboliai (išskyrus skyrybą)	re.sub(r"[^0-10a-zA-ZąčęėįšųūžĄČĘĖĮŠŲŪŽ.,?!:;\-\n\s]", "", text)
Pašalinama pasikartojanti struktūra gale	split("Taip pat skaitykite" / "Šaltiniai" / kt.)[:-1]
Pabaigoje – galutinis tarpo normalizavimas	re.sub(r"\s+", " ", text).strip()

Po valymo, tekstai buvo suskaidyti į sakinių blokus naudojant SpaCy *xx_ent_wiki_sm* modelį ir papildomą segmentavimo logiką, skiriančią lietuviškas santrumpas. Kiekvienas kontekstas buvo apribotas iki ~256 simbolių, kad būtų tinkamas naudoti generatyviniuose modeliuose.

Klausimai ir atsakymai buvo kuriami pasitelkiant LLM („Gemini 1.5 Flash“) per REST API, kiekvienam kontekstui generuojant 1–3 klausimų-atsakymų porų. Sugeneruoti atsakymai buvo automatiškai saugomi JSON formatu.

Atlikta apie 5 % sugeneruotų duomenų neformali peržiūra leidžia teigti, kad sugeneruotas turinys daugeliu atvejų yra logiškas ir tinkamas naudoti.

3.2.2. Verstas SQuAD duomenų rinkinys

Norint padidinti modelių priderinimo efektyvumą, tyrime buvo naudojamas anglų kalba sukurtas SQuAD (Stanford Question Answering Dataset) duomenų rinkinys. SQuAD rinkinys pasirinktas dėl aukštos kokybės ir aiškiai pateikiamų kontekstų, klausimų ir atsakymų grupių. Kiti anglų kalbos klausimų ir atsakymų duomenų rinkiniai, kaip Natural Questions [45] ar NewsQA [46], pasižymi sudėtingesne struktūra ir laisvesnės formos atsakymais, todėl jų pritaikymas verčiant tekstą būtų reikalavęs sudėtingesnių korekcijų ar papildomo anotavimo. SQuAD rinkinys yra vienas plačiausiai naudojamų šaltinių kontekstinių klausimų-atsakymų užduotims treniruoti ir testuoti. Jis sudarytas iš straipsnių, prie kurių pateikti žmogaus suformuluoti klausimai ir atitinkami atsakymai [25].

Iš kiekvieno SQuAD įrašo buvo ištraukti trys komponentai:
- kontekstas – pastraipa iš straipsnio,
- klausimas – tiesioginis klausimas, kurį galima atsakyti remiantis kontekstu,
- atsakymas – tekstas, tiesiogiai atsakantis į klausimą.

Visi šie elementai buvo automatiškai verčiami į lietuvių kalbą. Vertimas buvo atliekamas naudojant *googletrans* biblioteką, kuri pasitelkia „Google Translate“ paslauga.

3.2.3. Galutinis duomenų rinkinys

Galutinis bazinis duomenų rinkinys buvo sujungtas iš Wikipedia ir SQuAD šaltinių. Iš viso rinkinys susideda iš 78 731 treniravimo ir 8 748 testavimo įrašų. Kiekviename įraše pateikiamas teksto kontekstas, klausimas bei atsakymas, leidžiantys naudoti tą patį duomenų rinkinį tiek klausimų kūrimui, tiek atsakymų generavimui. Testavimo rinkinys sudaro 10% bendro įrašų kiekio, treniravimo rinkinys – 90%. Duomenų praturtinimui bei modelių treniravimui bendras rinkinys buvo sumažintas iki 20 000 įrašų, atitinkamai 18 000 – treniravimui, 2 000 – testavimui.

3.2.4. Praturtinimo metodai

Norint įvertinti duomenų kokybės įtaką klausimų kūrimo ir atsakymų generavimo užduotims, duomenų rinkiniai buvo papildomai praturtinti pasitelkiant automatizuotus metodus: atvirkštiniu vertimu, parafrazavimu ir sinonimų keitimu. Kiekvienas praturtinimo metodas buvo taikomas atskirai, o gauti duomenys naudoti modelių treniravimui.

Tyrime taikyti trys duomenų praturtinimo metodai.

Atvirkštinis vertimas

Tekstas išverčiamas į kitą kalbą (anglų), o tada atgal į lietuvių. Duomenų rinkinio kūrimui vertimai atlikti kontekstams bei klausimams. Atsakymams vertimas netaikomas, nes baziniame duomenų rinkinyje vyrauja trumpi atsakymai, kurie vertimo metu lengvai praranda prasmę. Vertimai atlikti automatiškai naudojant *googletrans* biblioteka.

Atlikus vertimą gauta 19 993 naujų įrašų. Likusieji 7 įrašai atmesti, nes jų vertimas sutapo su originalaus duomenų rinkinio tekstu. Sujungus naujus įrašus su baziniu duomenų rinkiniu gautas eksperimentams naudojamas duomenų rinkinys, kuriame iš viso 39 993 įrašų

Parafrazavimas

Parafrazavimas atliktas taikant generatyvųjį kalbos modelį Gemini Flash 2.0 su specialiais nurodymais (angl. *prompt*). Parafrazavimas taikytas tik klausimams, nes taikant šį metodą kontekstui, kyla rizika prarasti esminę informaciją, kuri naudojama atsakymui išgauti. Atsakymai baziniame duomenų rinkinyje yra pakankamai trumpi, todėl parafrazavimas nebūtų efektyvus.

Atlikus parafrazavimą gauta 19 607 naujų įrašų.

Sinonimų keitimas

Sinonimų keitimas taip pat atliktas taikant kalbos modelį Gemini Flash 2.0. Šiam metodui taikyti taip pat svarstytas taisyklėmis paremtas žodžių apkeitimas, tačiau dėl nepakankamų išteklių tyrime jis nebuvo taikomas. Metodas taikytas kontekstui, nes kontekstas yra daugiausiai teksto turinti duomenų rinkinio dalis, todėl jame yra didžiausia galimybė padidinti žodžių įvairovę. Kalbos modelio nurodymuose nustatyta, kad sugeneruotame tekste nebūtų keičiama esminė informacija, susijusi su klausimu bei atsakymu.

Atlikus sinonimų keitimą gauta 14 855 naujų įrašų.

Kiekvienas sugeneruotas rinkinys įrašomas atskirame faile. Taip užtikrinama kiekvieno metodo įtakos analizė, išvengiant kelių metodų tarpusavio sąveikos.

3.3. Modelių treniravimo metodika

Eksperimentinei daliai pasirinktas transformerių architektūros modelis mT5-small – tai – daugiakalbis modelis, kuris jau apmokytas su lietuvių kalbos duomenimis ir palaiko teksto generavimo užduotis. Šis modelis pasirinktas dėl šių priežasčių:

- Išankstinis treniravimas: mT5 modelis yra treniruotas su 101 kalba, įskaitant lietuvių, ir pasižymi gebėjimu atlikti įvairias teksto generavimo užduotis. Nors treniravimo metu mažiau

dėmesio skiriama mažoms kalboms, lietuvių kalba vis tiek įtraukta į originalius treniravimo duomenis.
- Modelio prieinamumas: mT5-small dėl savo dydžio yra tinkamas naudoti ribotus resursus turinčioje aplinkoje. Didesni modeliai dėl riboto VRAM ir treniravimo laiko apribojimų nebuvo praktiškai tinkami šiam tyrimui.
- Architektūra: modelis remiasi encoder–decoder architektūra, kuris leidžia vienodai efektyviai naudoti tiek klausimų kūrimo, tiek atsakymų generavimo užduotims. Tai suteikia galimybę taikyti vienodą metodiką abiems užduotims.

Taip pat svarstyti modeliai, iš anksto priderinti lietuviškiems tekstams (LT-Llama-2, t5-base-lithuanian-news-summaries-17, ByT5-Lithuanian-gec-100h), bet pasirinkta jų nenaudoti dėl didesnių techninių resursų reikalavimų. mT5-small pasirinktas kaip tinkamiausias kompromisas tarp našumo ir kalbos palaikymo.

3.3.1. Modelio versijos

Eksperimento metu lyginami mT5-small modelio treniravimo variantai:
1. Originalus mT5-small modelis, be papildomo priderinimo – naudojamas kaip atskaitos taškas.
2. mT5-small modelis, treniruotas su lietuvišku klausimų-atsakymų rinkiniu.
3. mT5-small modelis, treniruotas su įvairiais praturtintais duomenų rinkiniais.

3.3.2. Mokymo architektūra ir sąlygos

Modelių treniravimas atliktas pasitelkiant PyTorch pagrindu sukurtą „transformers“ biblioteką. Duomenų paruošimas vykdytas taip, kad kiekvienas įrašas turėtų įvestį, atitinkančią vieną iš šių formų:
- Klausimų generavimui: generate question: {kontekstas}
- Atsakymų generavimui: question: {klausimas} context: {kontekstas}

Tokiu būdu duomenys pateikiami kaip tekstinė užklausa, kurios formatas tinkamas Text-to-Text architektūros modeliams. Tai leidžia vienodai apdoroti įvairias užduotis kaip teksto generavimo užklausas. Modelis mokosi susieti tokio formato užklausą su tikėtina išvestimi.

Visų modelių mokymo metu naudoti bendri fiksuoti parametrai (žr. 10 lentelė.)

10 lentelė. Modelio treniravimo nustatymai ir hiperparametrai

Parametras	**Reikšmė**
Modelio architektūra	mT5-small
Optimizatorius	AdamW
Epochų skaičius	2
Mokymosi tempas (lr)	5.00E-05
Batch size	64
Gradient accumulation	1 (keičiamas, jei trūksta atminties resursų treniruojant modelį, kad būtų pasiekiamas efektyvus 64 *batch_size*)
Maks. konteksto ilgis	512
Maks. išvesties ilgis	64

Patikrinimo intervalas	kas 500 žingsnių

3.3.3. Techniniai apribojimai

Modelių treniravimas ir eksperimentai buvo vykdomi nuotolinėje *RunPod* debesijos aplinkoje, pasirinktoje dėl galimybės naudoti didesnės skaičiavimo galios vaizdo plokštę. Tokios platformos leidžia kurti ir vykdyti individualiai konfigūruotus konteinerius, skirtus mašininio mokymosi užduotims.

Eksperimentuose naudota sistema pasižymėjo šiomis specifikacijomis:
- GPU: 1 NVIDIA RTX 4000 Ada (20 GB VRAM)
- CPU: 16 vCPU
- Operatyvioji atmintis (RAM): 62 GB
- Aplinka: konteinerizuota Ubuntu sistema

Ši GPU architektūra yra pakankamai nauja ir leidžia efektyviai dirbti su sekos į seką architektūros modeliais (angl. *sequence-to-sequence*). 20 GB VRAM leidžia naudoti didesnį *batch_size* parametrą, GPU resursai naudojami tiesioginiam tensorių skaičiavimų spartinimui. Net su pažangia aplinka, tyrimas vis tiek susiduria su kai kuriais ribojimais:
- Modeliai, didesni nei mT5-small, reikalautų >24 GB VRAM, arba jų apmokymas užtruktų virš 12 valandų per modelį, todėl mT5-base eksperimentai buvo techniškai neprieinami.
- Dėl laiko sąnaudų kiekvienas modelis turi būti treniruojamas ribotu epochų skaičiumi.

3.4. Modelių vertinimo kriterijai

Modelių veikimo kokybei vertinti šio tyrimo metu taikomos automatinės teksto generavimo metrikos – BLEU ir ROUGE, kurios leidžia objektyviai palyginti sugeneruotą klausimą arba atsakymą su žinomais tiksliniais duomenimis.

Vertinimas vykdomas dvejuose atskiruose kontekstuose:
- Klausimų kūrimo užduotyje – kiek sugeneruotas klausimas sutampa su tiksliniu klausimu.
- Atsakymų generavimo užduotyje – kiek sugeneruotas atsakymas atitinka tikslinį atsakymą.

Modeliai testuojami dviem atskirais duomenų pogrupiais:
1. Treniravimo dalis: vertinama su duomenimis, kurie buvo naudoti modelio treniravimo etape.
2. Testavimo dalis: vertinama, kaip modelis apdoroja nematytus kontekstus.

Sugeneruoti rezultatai vertinami pagal šias metrikas:
- BLEU-1 iki BLEU-4 – kiek n-gramų sugeneruotoje išvestyje randamos tiksliniame tekste.
- ROUGE-1, ROUGE-2, ROUGE-L – kiek n-gramų tiksliniame tekste yra sugeneruotoje išvestyje.

3.5. Tyrimo eiga

Šiame poskyryje pateikiamas planas, pagal kurį vertinami skirtingi modelių treniravimo variantai klausimų kūrimo ir atsakymų generavimo užduotyse lietuvių kalba. Klausimų generavimo ir atsakymo modeliai treniruojami ir vertinami atskirai, naudojant skirtingas duomenų rinkinių versijas. Naudojamas mT5-small modelis, kuris yra papildomai treniruojamas skirtingais duomenų rinkiniais. Duomenų rinkinių sudarymas yra aprašytas 3.2 poskyryje. Kiekvienam eksperimentui naudojamas

skirtingu būdu praturtintas duomenų rinkinys. Išskiriamos šios eksperimentinės grupės (žr. 11 lentelė).

11 lentelė. Eksperimentų grupės

Modelis	**Paskirtis**	**Duomenų rinkinys**	**Praturtinimo metodas**
Q0	QG	—	—
Q1	QG	Bazinis (20 000)	—
Q2	QG	Didesnis (40 000)	—
Q3	QG	Bazinis	Atvirkštinis vertimas
Q4	QG	Bazinis	Parafrazavimas
Q5	QG	Bazinis	Sinonimų keitimas
A0	QA	—	—
A1	QA	Bazinis (20 000)	—
A2	QA	Didesnis (40 000)	—
A3	QA	Bazinis	Atvirkštinis vertimas
A4	QA	Bazinis	Parafrazavimas
A5	QA	Bazinis	Sinonimų keitimas

Kiekvienam iš šių variantų naudojamas atskirai apmokytas modelis. Tai leidžia įvertinti kiekvieno duomenų praturtinimo metodo įtaką rezultatų kokybei. Be to, kiekvienai užduočiai pateikiama bazinė modelio versija (Q0 ir A0), kuri leidžia įvertinti modelio priderinimo naudą net be papildomo praturtinimo.

Modelių vertinimas atliekamas pagal BLEU (1–4) ir ROUGE (1, 2, L) metrikas, aprašytas 3.4 poskyryje. Testavimui naudojamas testavimo rinkinys, kuriame esantys duomenys nebuvo naudoti treniravimui.

4. Eksperimentinė dalis

Šiame skyriuje analizuojami modelių veikimo rezultatai, gauti atliekant klausimų kūrimo ir atsakymų generavimo užduotis. Vertinimui pagrinde taikytos BLEU ir ROUGE metrikos, leidžiančios objektyviai įvertinti sugeneruoto teksto panašumą į tikslinį rezultatą.

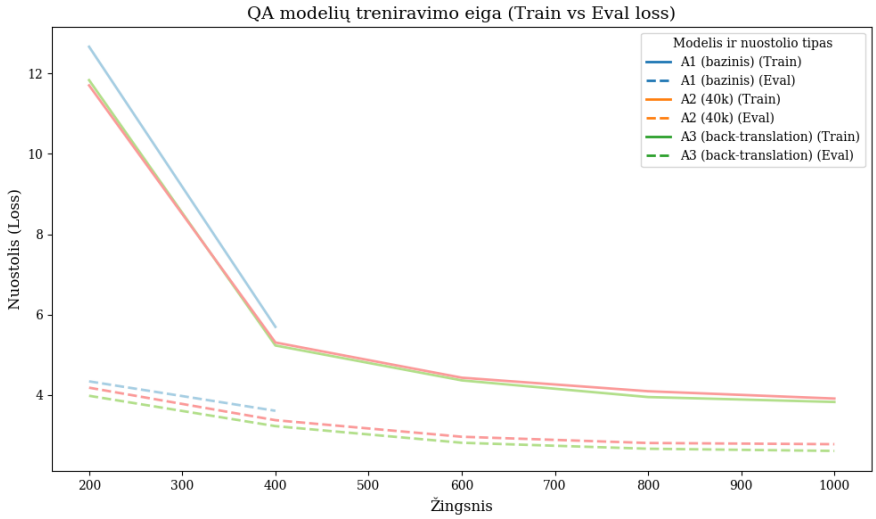
Modelių veikimas vertintas abiem duomenų pogrupiais – treniravimo (*train*) ir testavimo (*test*) – siekiant įvertinti modelio gebėjimus ne tik matytame kontekste, bet ir naujame.

4.1. Tyrimo rezultatai

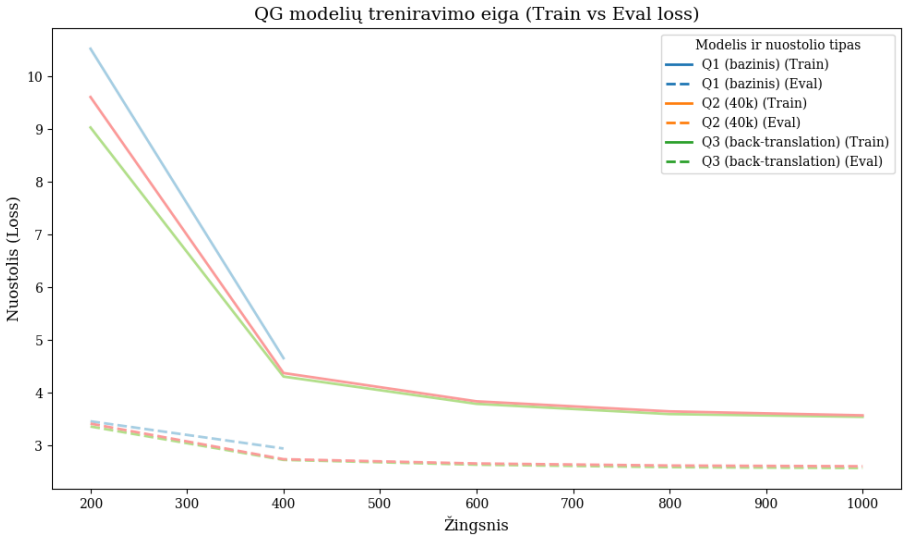
4.1.1. Treniravimo eiga

Modelių treniravimui buvo naudojami vienodi hiperparametrai (žr. 3.5 skyrių), siekiant užtikrinti rezultatų palyginamumą. Kiekvieno modelio treniravimo metu buvo registruojamos *train_loss, eval_loss, eval_bleu, eval_rougeL* metrikos, leidžiančios įvertinti mokymosi eigą.

Kaip matyti iš 17 pav. ir 18 pav., atsakymų generavimo modelių nuostolis mažėja nuosekliai visų modelių atveju.



17 pav. Klausimų atsakymo modelių treniravimo eiga



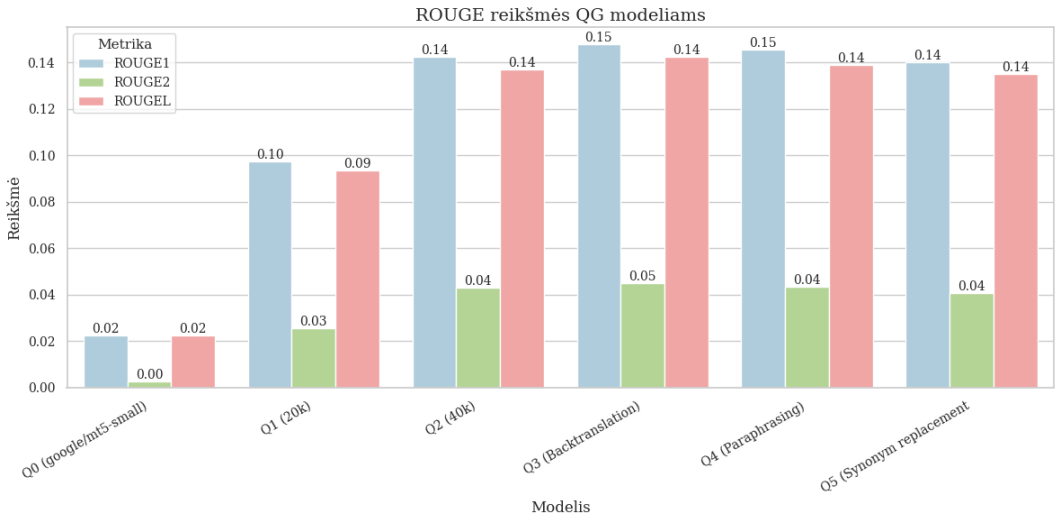
18 pav. Klausimų generavimo modelių treniravimo eiga

4.1.2. Klausimų kūrimo modelių rezultatai

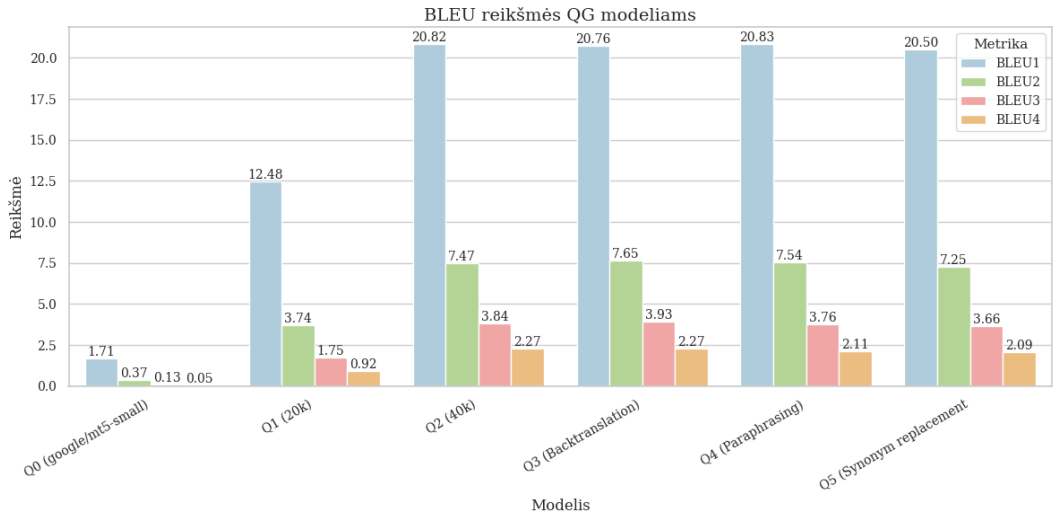
Klausimų kūrimo modelių vertinimo rezultatai pateikiami 12 lentelėje. Kiekvienas modelis buvo vertinamas pagal BLEU (1-4) ir ROUGE (1, 2, L) metrikas. Vizualinis šių metrikų palyginimas pateikiamas 19 pav. ir 20 pav.

12 lentelė. Klausimų kūrimo metrikų palyginimas galutiniuose modeliuose

Eksperimentas	**Treniravimo duomenys**	**BLEU-1**	**BLEU-4**	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**
Q0	–	1.7145	0.0549	0.0224	0.0026	0.0223
Q1	Bazinis (20 000)	12.4763	0.9208	0.0974	0.0256	0.0935
Q2	Bazinis (40 000)	20.8220	2.2709	0.1425	0.0430	0.1372
Q3	Praturtintas (atvirkštinis vertimas)	20.7582	2.2724	0.1477	0.0450	0.1425
Q4	Praturtintas (parafrazavimas)	20.8260	2.1103	0.1455	0.0436	0.1391
Q5	Praturtintas (sinonimų keitimas)	20.5011	2.0927	0.1401	0.0405	0.1350



19 pav. ROUGE metrikų palyginimas klausimų generavimo modeliams



20 pav. BLEU metrikų palyginimas klausimų generavimo modeliams

Kaip matyti iš lentelės ir grafikų, visi modeliai, sukurti naudojant praturtintus duomenų rinkinius (Q3–Q5) pasiekė aukštesnius rezultatus nei bazinis 20k modelis (Q1):

- BLEU-4 reikšmė padidėjo nuo 0,92 (Q1) iki 2,27 (su atvirkštinio vertimo metodu),
- ROUGE-L padidėjo nuo 0,09 iki daugiau nei 0,14.

Svarbiausias pastebėjimas – kai kurie duomenų praturtinimo modeliai pranoko ne tik bazinį (Q1), bet ir su 40 000 originalių įrašų treniruotą modelį (Q2). Tai rodo praturtinimo metodų efektyvumą klausimų generavimo užduotyse.

Tarp visų eksperimentų:

- Atvirkštinio vertimo (Q3) ir parafrazavimo (Q4) modeliai pasirodė geriausiai.

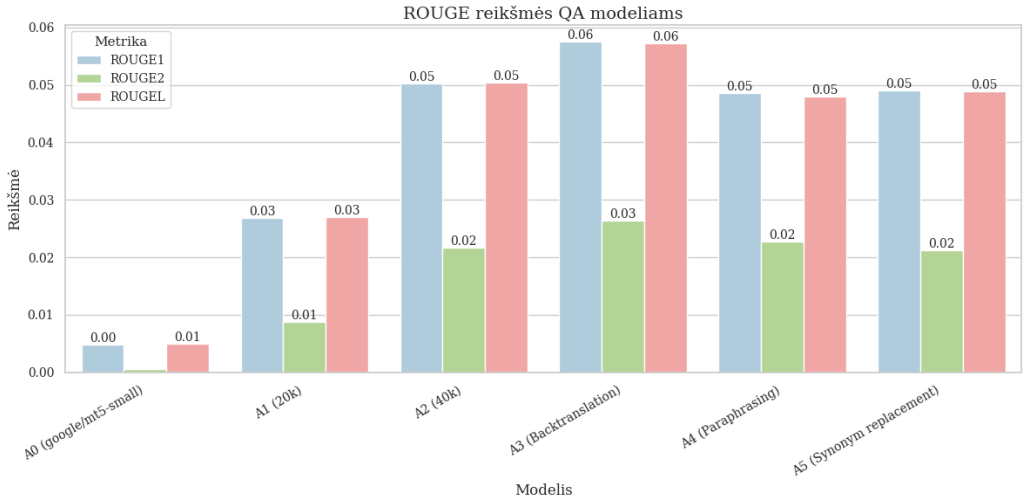
– Sinonimų keitimas (Q5) taip pat viršijo bazinio modelio (Q1) rezultatus, bet pasirodė blogiau už kitus duomenų praturtinimo metodus ir modelio treniravimą su daugiau originalių įrašų.

4.1.3. Atsakymų modelių rezultatai

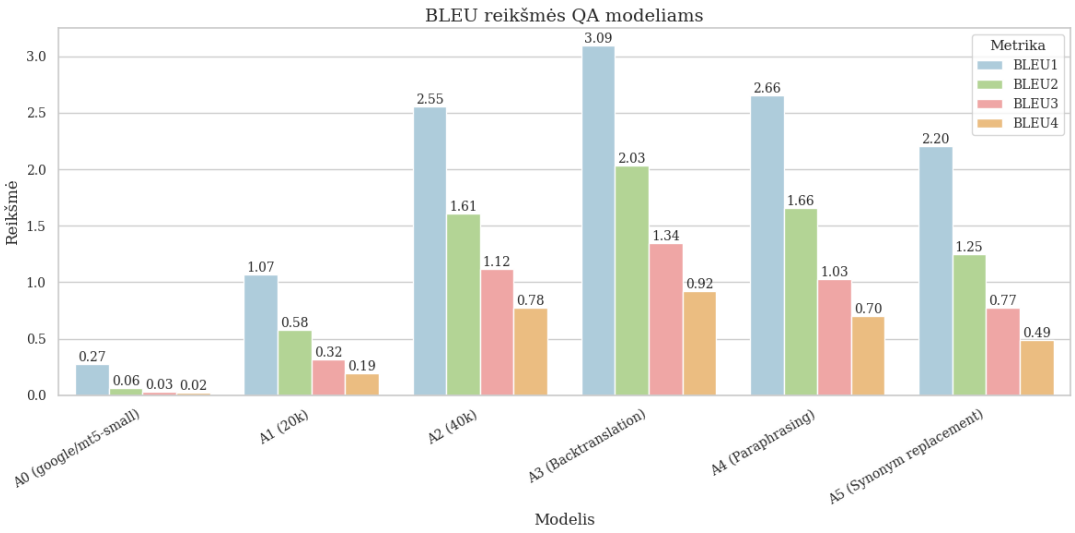
Atsakymų generavimo modelių metrikos pateikiamos 13 lentelėje, o BLEU ir ROUGE metrikų palyginimas – 21 pav. ir 22 pav.

13 lentelė. Klausimų atsakymo metrikų palyginimas galutiniuose modeliuose

Eksperimentas	**Treniravimo duomenys**	**BLEU-1**	**BLEU-4**	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**
A0	–	0.2739	0.0191	0.0048	0.0006	0.0050
A1	Bazinis (20 000)	1.066	0.1924	0.0269	0.0087	0.0269
A2	Bazinis (40 000)	2.552	0.7775	0.0502	0.0216	0.0503
A3	Praturtintas (atvirkštinis vertimas)	3.093	0.922	0.0574	0.0264	0.0571
A4	Praturtintas (parafrazavimas)	2.658	0.698	0.0485	0.0227	0.0480
A5	Praturtintas (sinonimų keitimas)	2.2010	0.4862	0.0490	0.0211	0.0488



21 pav. ROUGE metrikų palyginimas klausimų atsakymo modeliams



22 pav. BLEU metrikų palyginimas klausimų atsakymo modeliams

Rezultatai rodo tą pačią tendenciją kaip ir klausimų kūrimo užduotyje:

- BLEU-4 išauga nuo 0,19 (A1, bazinis) iki 0,92 (A3),
- ROUGE-L padidėja nuo 0,027 iki 0,057.

Ypač reikšminga tai, kad atvirkštinio vertimo (A3) modelis lenkia 40 000 originalių įrašų modelį (A2) visose metrikose. Šie rezultatai parodo, kad praturtinimo suteikiama duomenų semantinė įvairovė pagerina modelio gebėjimą apibendrinti informaciją.

4.2. Tyrimo apribojimai

Tyrimo metu susidurta su keliais svarbiais apribojimais, kurie turėjo įtaką rezultatų gavimui:

- Kalbos ribotumas – lietuvių kalba pasižymi ribotais ištekliais, todėl net ir pažangūs modeliai, tokie kaip mT5, ne visada geba generuoti rišlius ar semantiškai tikslius klausimus bei atsakymus.
- Duomenų kokybė – duomenų rinkinio didžiąją dalį sudaro SQuAD rinkinys, kuris buvo išverstas automatiškai, dėl to modelių generuojami tekstai gali turėti trūkumų.
- Modelio architektūra: Dėl skaičiavimo išteklių buvo pasirinktas mT5-small modelis. Didesni modeliai (pvz., mT5-base, mT5-large) galėjo pateikti geresnius rezultatus, tačiau jie nebuvo prieinami dėl techninių apribojimų.
- Metrikų ribotumas – automatinės metrikos (BLEU, ROUGE) nesugeba visapusiškai įvertinti semantinio nuoseklumo ar loginio atsakymo atitikimo. Rankinis vertinimas galėjo papildyti analizę, bet nebuvo įtrauktas dėl praktinių apribojimų.
- Treniravimo ribos – dėl techninių resursų apribojimų kiekvienas modelis treniruotas fiksuotu epochų skaičiumi, o tai galėjo riboti optimalaus apmokymo pasiekimą.

4.3. Tyrimo išvados

Šio tyrimo tikslas buvo ištirti duomenų praturtinimo įtaką mašininio mokymosi modelių gebėjimui generuoti ir atsakyti į klausimus lietuvių kalba. Pagrindinės išvados:

1. Duomenų praturtinimas ženkliai pagerino modelių kokybę, lyginant su modeliais, treniruotais tik su baziniu 20 000 įrašų rinkiniu. Visose metrikose buvo užfiksuotas pastebimas augimas ir

klausimų atsakymo, ir klausimų generavimo užduotyse. BLEU-4 reikšmės klausimų kūrimo modelyje padidėjo nuo 0,92 (Q1) iki 2,27 (Q3), o atsakymų generavimo modelyje – nuo 0,19 (A1) iki 0,92 (A3).

2. Atvirkštinio vertimo ir parafrazavimo metodai parodė didžiausią naudą – tiek klausimų kūrimo, tiek atsakymų generavimo užduotyse. BLEU-4 reikšmės šiuose modeliuose buvo daugiau nei 2 kartus didesnės nei bazinio modelio.
3. Modeliai, treniruoti naudojant praturtintus duomenų rinkinius, nors nežymiai, bet pasiekė geresnius rezultatus nei modeliai, treniruoti su atitinkamu skaičiumi originalių įrašų. Tai leidžia daryti prielaidą, kad semantinė ir struktūrinė duomenų įvairovė turi didelę įtaką rezultatams.
4. Sinonimų keitimo metodas, nors ir buvo mažiausiai efektyvus iš trijų taikytų, vis tiek pastebimai pagerino rezultatus lyginant su baziniu modeliu. BLEU-4 reikšmė kilo nuo 0,92 iki 2,09 klausimų generavimo užduotyje ir nuo 0,19 iki 0,48 klausimų atsakymo užduotyje. Šio metodo efektyvumas galėjo būti mažiau efektyvus dėl mažesnio papildomo duomenų kiekio.
5. Gauti rezultatai rodo, kad net esant ribotiems resursams, galima ištreniruoti aukštos kokybės teksto generavimo modelius naudojant automatinio duomenų praturtinimo metodus.
6. Tyrimas patvirtino, kad duomenų kokybė ir įvairovė mažai išteklių turinčiose kalbose (kaip lietuvių) yra ypač svarbūs aspektai treniruojant tekstą generuojančius modelius. Automatiniai transformacijos metodai, taikomi kontroliuojant semantinius teksto pokyčius, gali būti efektyvus būdas pagerinti teksto generavimo modelių veikimą.

Išvados

1. Darbe buvo išanalizuoti tekstą generuojantys kalbos modeliai ir jų taikymas klausimų kūrimo bei atsakymų generavimo užduotims. Nustatyta, kad mažai resursų turinčiose kalbose šie modeliai neveikia pakankamai tiksliai be papildomo priderinimo.
2. Buvo sukurtas duomenų rinkinys, apimantis verstą SQuAD rinkinį ir tekstus iš Vikipedijos. Šis duomenų rinkinys buvo priderintas klausimų ir atsakymų generavimui lietuvių kalba.
3. Duomenų praturtinimui naudoti trys metodai: atvirkštinis vertimas, parafrazavimas ir sinonimų keitimas. Kiekvienas metodas buvo taikomas atskirai ir vertinamas pagal poveikį modelių kokybei.
4. Tyrime naudotas mT5-small modelis, kuris buvo treniruotas klausimų kūrimui ir atsakymų generavimui. Modeliai buvo vertinami pagal BLEU ir ROUGE metrikas.
5. Visi modeliai, treniruoti su praturtintais duomenimis veikė geriau nei modeliai treniruoti su tik baziniu duomenų rinkiniu. Modelių, sukurtų naudojant atvirkštinio vertimo metodu praturtintu duomenų rinkiniu, metrikos viršijo net modelių, kurie buvo treniruoti naudojant 40 000 originalių įrašų. Tai rodo duomenų praturtinimo metodų efektyvumą treniruojant kalbos modelius mažai resursų turinčiomis kalbomis.
6. Geriausi rezultatai buvo pasiekti naudojant duomenų rinkinį, praturtintą atvirkštinio vertimo metodu – su juo treniruotų modelių BLEU-4 ir ROUGE metrikos buvo aukščiausios abiejose užduotyse. Lyginant su baziniu duomenų rinkiniu BLEU4 metrika klausimų generavimo užduotyje padidėjo nuo 0,19 iki 0,92, o klausimų atsakymo – nuo 0,92 iki 2,27.
7. Gauti rezultatai rodo, kad net riboti resursai gali būti panaudoti efektyviai treniruojant kalbos modelius, jei taikomos automatinės duomenų praturtinimo technikos.

Literatūros sąrašas

1. GUO, Shasha. Lizi LIAO. Cuiping LI. ir kt. A Survey on Neural Question Generation: Methods, Applications, and Prospects. *Proceedings of the Thirty-ThirdInternational Joint Conference on Artificial Intelligence* [interaktyvus]. Jeju, South Korea: International Joint Conferences on Artificial Intelligence Organization, 2024, p. 8038–8047. [žiūrėta 2025-05-15]. Prieiga per: https://www.ijcai.org/proceedings/2024/889.
2. PIRYANI, Bhawna. Jamshid MOZAFARI. ir Adam JATOWT. ChroniclingAmericaQA: A Large-scale Question Answering Dataset based on Historical American Newspaper Pages. *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval* [interaktyvus]. 2024, p. 2038–2048. [žiūrėta 2025-05-15]. Prieiga per: http://arxiv.org/abs/2403.17859.
3. DU, Xinya. Junru SHAO. ir Claire CARDIE. [interaktyvus]. [s.l.]: arXiv, 2017 [žiūrėta 2025-05-15]. arXiv:1705.00106 [cs]. Prieiga per: http://arxiv.org/abs/1705.00106.
4. RAFFEL, Colin. Noam SHAZEER. Adam ROBERTS. ir kt. [interaktyvus]. [s.l.]: arXiv, 2023 [žiūrėta 2025-05-15]. arXiv:1910.10683 [cs]. Prieiga per: http://arxiv.org/abs/1910.10683.
5. LEWIS, Mike. Yinhan LIU. Naman GOYAL. ir kt. [interaktyvus]. [s.l.]: arXiv, 2019 [žiūrėta 2025-05-15]. arXiv:1910.13461 [cs]. Prieiga per: http://arxiv.org/abs/1910.13461.
6. NGUYEN, Minh-Tien. Nhung BUI. Manh TRAN-TIEN. ir kt. [interaktyvus]. [s.l.]: arXiv, 2024 [žiūrėta 2025-05-15]. arXiv:2212.12192 [cs]. Prieiga per: http://arxiv.org/abs/2212.12192.
7. XUE, Linting. Noah CONSTANT. Adam ROBERTS. ir kt. [interaktyvus]. [s.l.]: arXiv, 2021 [žiūrėta 2025-05-15]. arXiv:2010.11934 [cs]. Prieiga per: http://arxiv.org/abs/2010.11934.
8. CLARK, Jonathan H. Eunsol CHOI. Michael COLLINS. ir kt. T Y D I QA: A Benchmark for Information-Seeking Question Answering in *Ty* pologically *Di* verse Languages. *Transactions of the Association for Computational Linguistics*. 2020, Vol. 8, p. 454–470.
9. MCGIFF, Josh. ir Nikola S. NIKOLOV. [interaktyvus]. [s.l.]: arXiv, 2025 [žiūrėta 2025-05-15]. arXiv:2505.04531 [cs]. Prieiga per: http://arxiv.org/abs/2505.04531.
10. ULČAR, Matej. ir Marko ROBNIK-ŠIKONJA. LitLat BERT. *https://embeddia.eu* [interaktyvus]. 2020, [žiūrėta 2025-05-15]. Prieiga per: https://clarin.vdu.lt/xmlui/handle/20.500.11821/42.
11. NAKVOSAS, Artūras. Povilas DANIUŠIS. ir Vytas MULEVIČIUS. Open Llama2 Model for the Lithuanian Language. *Informatica*. 2025, p. 1–22.
12. PEREVALOV, Aleksandr. Dennis DIEFENBACH. Ricardo USBECK. ir kt. [interaktyvus]. [s.l.]: arXiv, 2022 [žiūrėta 2025-05-15]. arXiv:2202.00120 [cs]. Prieiga per: http://arxiv.org/abs/2202.00120.
13. FENG, Steven. Varun GANGAL. Jason WEI. ir kt. A Survey of Data Augmentation Approaches for NLP. *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021* [interaktyvus]. Online: Association for Computational Linguistics, 2021, p. 968–988. [žiūrėta 2025-05-15]. Prieiga per: https://aclanthology.org/2021.findings-acl.84.
14. LEWIS, Patrick. Ludovic DENOYER. ir Sebastian RIEDEL. Unsupervised Question Answering by Cloze Translation. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* [interaktyvus]. 2019, p. 4896–4910. [žiūrėta 2025-05-15]. Prieiga per: http://arxiv.org/abs/1906.04980.
15. PURASON, Taido. Hele-Andra KUULMETS. ir Mark FISHEL. [interaktyvus]. [s.l.]: arXiv, 2025 [žiūrėta 2025-05-15]. arXiv:2410.18902 [cs]. Prieiga per: http://arxiv.org/abs/2410.18902.
16. PETRAUSKAITĖ, Rūta. Darius AMILEVIČIUS. Virginijus DADURKEVIČIUS. ir kt. CLARIN-LT: Home for Lithuanian Language Resources. FIŠER, D. - WITT, A.Sud. *CLARIN* [interaktyvus]. [s.l.]: De Gruyter, 2022, p. 511–536. [žiūrėta 2025-05-15]. ISBN 978-3-11-076737-7Prieiga per: https://www.degruyter.com/document/doi/10.1515/9783110767377-020/html.

17. LIU, Yinhan. Jiatao GU. Naman GOYAL. ir kt. [interaktyvus]. [s.l.]: arXiv, 2020 [žiūrėta 2025-05-15]. arXiv:2001.08210 [cs]. Prieiga per: http://arxiv.org/abs/2001.08210.

18. TOUVRON, Hugo. Thibaut LAVRIL. Gautier IZACARD. ir kt. [interaktyvus]. [s.l.]: arXiv, 2023 [žiūrėta 2025-05-15]. arXiv:2302.13971 [cs]. Prieiga per: http://arxiv.org/abs/2302.13971.

19. STANKEVIČIUS, Lukas. ir Mantas LUKOŠEVIČIUS. Generating Abstractive Summaries of Lithuanian News Articles Using a Transformer Model. LOPATA, A. - GUDONIENĖ, D. - BUTKIENĖ, R.Sud. *Information and Software Technologies*. Cham: Springer International Publishing, 2021, p. 341–352.

20. STANKEVIČIUS, Lukas. ir Mantas LUKOŠEVIČIUS. Towards Lithuanian Grammatical Error Correction. SILHAVY, R.Sud. *Artificial Intelligence Trends in Systems*. Cham: Springer International Publishing, 2022, p. 490–503.

21. WANG, Bingning. Ting YAO. Weipeng CHEN. ir kt. Multi-Lingual Question Generation with Language Agnostic Language Model. *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021* [interaktyvus]. Online: Association for Computational Linguistics, 2021, p. 2262–2272. [žiūrėta 2025-05-15]. Prieiga per: https://aclanthology.org/2021.findings-acl.199.

22. ALBERTI, Chris. Daniel ANDOR. Emily PITLER. ir kt. [interaktyvus]. [s.l.]: arXiv, 2019 [žiūrėta 2025-05-15]. arXiv:1906.05416 [cs]. Prieiga per: http://arxiv.org/abs/1906.05416.

23. AGRAWAL, Priyanka. Chris ALBERTI. Fantine HUOT. ir kt. QAMELEON : Multilingual QA with Only 5 Examples. *Transactions of the Association for Computational Linguistics*. 2023, Vol. 11, p. 1754–1771.

24. ZHA, Daochen. Zaid Pervaiz BHAT. Kwei-Herng LAI. ir kt. [interaktyvus]. [s.l.]: arXiv, 2023 [žiūrėta 2025-05-17]. arXiv:2303.10158 [cs]. Prieiga per: http://arxiv.org/abs/2303.10158.

25. RAJPURKAR, Pranav. Jian ZHANG. Konstantin LOPYREV. ir kt. [interaktyvus]. [s.l.]: arXiv, 2016 [žiūrėta 2025-05-15]. arXiv:1606.05250 [cs]. Prieiga per: http://arxiv.org/abs/1606.05250.

26. KURDI, Ghader. Jared LEO. Bijan PARSIA. ir kt. A Systematic Review of Automatic Question Generation for Educational Purposes. *International Journal of Artificial Intelligence in Education*. 2020, Vol. 30, no. 1, p. 121–204.

27. BUDUR, Emrah. Rıza ÖZÇELIK. Dilara SOYLU. ir kt. Building Efficient and Effective OpenQA Systems for Low-Resource Languages. *Knowledge-Based Systems*. 2024, Vol. 302, p. 112243.

28. MORENO-CEDIEL, Antonio. Jesus-Angel DEL-HOYO-GABALDON. Eva GARCIA-LOPEZ. ir kt. Evaluating the performance of multilingual models in answer extraction and question generation. *Scientific Reports*. 2024, Vol. 14, no. 1, p. 15477.

29. CARRINO, Casimiro Pio. Marta R. COSTA-JUSSÀ. ir José A. R. FONOLLOSA. [interaktyvus]. [s.l.]: arXiv, 2019 [žiūrėta 2025-05-17]. arXiv:1912.05200 [cs]. Prieiga per: http://arxiv.org/abs/1912.05200.

30. ABADANI, Negin. Jamshid MOZAFARI. Afsaneh FATEMI. ir kt. ParSQuAD: Persian Question Answering Dataset based on Machine Translation of SQuAD 2.0. *International Journal of Web Research* [interaktyvus]. 2021, Vol. 4, no. 1. [žiūrėta 2025-05-17]. Prieiga per: https://doi.org/10.22133/ijwr.2021.293313.1101.

31. PURI, Raul. Ryan SPRING. Mostofa PATWARY. ir kt. [interaktyvus]. [s.l.]: arXiv, 2020 [žiūrėta 2025-05-17]. arXiv:2002.09599 [cs]. Prieiga per: http://arxiv.org/abs/2002.09599.

32. SHAKERI, Siamak. Noah CONSTANT. Mihir KALE. ir kt. Towards Zero-Shot Multilingual Synthetic Question and Answer Generation for Cross-Lingual Reading Comprehension. *Proceedings of the 14th International Conference on Natural Language Generation* [interaktyvus]. Aberdeen, Scotland, UK: Association for Computational Linguistics, 2021, p. 35–45. [žiūrėta 2025-05-15]. Prieiga per: https://aclanthology.org/2021.inlg-1.4.

33. DING, Bosheng. Chengwei QIN. Ruochen ZHAO. ir kt. [interaktyvus]. [s.l.]: arXiv, 2024 [žiūrėta 2025-05-17]. arXiv:2403.02990 [cs]. Prieiga per: http://arxiv.org/abs/2403.02990.

34. TAKAHASHI, Kosuke. Takahiro OMI. Kosuke ARIMA. ir kt. Training Generative Question-Answering on Synthetic Data Obtained from an Instruct-tuned Model. . .

35. NAMBOORI, Amani. Shivam MANGALE. Andy ROSENBAUM. ir kt. [interaktyvus]. [s.l.]: arXiv, 2024 [žiūrėta 2025-05-15]. arXiv:2404.09163 [cs]. Prieiga per: http://arxiv.org/abs/2404.09163.

36. SENNRICH, Rico. Barry HADDOW. ir Alexandra BIRCH. [interaktyvus]. [s.l.]: arXiv, 2016 [žiūrėta 2025-05-15]. arXiv:1511.06709 [cs]. Prieiga per: http://arxiv.org/abs/1511.06709.

37. ZHOU, Jianing. ir Suma BHAT. Paraphrase Generation: A Survey of the State of the Art. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* [interaktyvus]. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021, p. 5075–5086. [žiūrėta 2025-05-15]. Prieiga per: https://aclanthology.org/2021.emnlp-main.414.

38. WEI, Jason. ir Kai ZOU. [interaktyvus]. [s.l.]: arXiv, 2019 [žiūrėta 2025-05-15]. arXiv:1901.11196 [cs]. Prieiga per: http://arxiv.org/abs/1901.11196.

39. PAPINENI, Kishore. Salim ROUKOS. Todd WARD. ir kt. BLEU: a method for automatic evaluation of machine translation. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02* [interaktyvus]. Philadelphia, Pennsylvania: Association for Computational Linguistics, 2001, p. 311. [žiūrėta 2025-05-15]. Prieiga per: http://portal.acm.org/citation.cfm?doid=1073083.1073135.

40. LIN, Chin-Yew. ROUGE: A Package for Automatic Evaluation of Summaries. . .

41. A. BALDHA, Nirav. Question Answering for Low Resource Languages Using Natural Language Processing. *International Journal of Scientific Research and Engineering Trends*. 2022, Vol. 8, no. 2, p. 1122–1126.

42. MOHAMMADSHAHI, Alireza. Thomas SCIALOM. Majid YAZDANI. ir kt. RQUGE: Reference-Free Metric for Evaluating Question Generation by Answering the Question. *Findings of the Association for Computational Linguistics: ACL 2023* [interaktyvus]. Toronto, Canada: Association for Computational Linguistics, 2023, p. 6845–6867. [žiūrėta 2025-05-15]. Prieiga per: https://aclanthology.org/2023.findings-acl.428.

43. ARTETXE, Mikel. Sebastian RUDER. ir Dani YOGATAMA. On the Cross-lingual Transferability of Monolingual Representations. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* [interaktyvus]. 2020, p. 4623–4637. [žiūrėta 2025-05-15]. Prieiga per: http://arxiv.org/abs/1910.11856.

44. KYLLIÄINEN, Ilmari. ir Roman YANGARBER. [interaktyvus]. [s.l.]: arXiv, 2022 [žiūrėta 2025-05-15]. arXiv:2211.13794 [cs]. Prieiga per: http://arxiv.org/abs/2211.13794.

45. KWIATKOWSKI, Tom. Jennimaria PALOMAKI. Olivia REDFIELD. ir kt. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics*. 2019, Vol. 7, p. 453–466.

46. TRISCHLER, Adam. Tong WANG. Xingdi YUAN. ir kt. [interaktyvus]. [s.l.]: arXiv, 2017 [žiūrėta 2025-05-25]. arXiv:1611.09830 [cs]. Prieiga per: http://arxiv.org/abs/1611.09830.

Priedai

1 priedas. Modelių metrikos

Modelis	BLEU	BLEU1	BLEU2	BLEU3	BLEU4	ROUGE1	ROUGE2	ROUGEL	Paskirtis
A0 (google/mt5-small)	0.01914087306	0.2739726027	0.06086736184	0.03071138985	0.01914087306	0.004885457546	0.000603846153 8	0.005003489025	qa
A1 (20k)	0.1924716411	1.07E+00	0.5783305999	0.3171824167	0.1924716411	0.02690069859	0.008740054828	0.02696342046	qa
A2 (40k)	0.7775886943	2.552648373	1.60932657	1.121275308	0.7775886943	0.05027393458	0.02163138944	0.05036006046	qa
A3 (Atvirkštinis vertimas)	0.9226470427	3.093843149	2.030514545	1.343702416	9.23E-01	0.05748669476	0.02641402976	0.05719274974	qa
A4 (Parafrazavimas)	0.6985502183	2.658326818	1.659144913	1.031317625	0.6985502183	0.04851867288	0.02270878394	0.04800663217	qa
A5 (Sinonimų keitimas)	0.4862443585	2.201096892	1.25092186	0.77322865	0.4862443585	0.04903709057	0.02116293373	0.04886973785	qa
Q0 (google/mt5-small)	0.05492964392	1.714577128	0.3673693908	0.1268418108	0.05492964392	0.02249310709	0.00267363992	0.02239100077	qg
Q1 (20k)	0.9208339794	12.476356306	3.738825935	1.750109141	0.9208339794	0.09742855742	0.02565678249	0.09355508387	qg
Q2 (40k)	2.270983833	20.822079	7.465888342	3.841793517	2.270983833	0.1425465294	0.04309345074	0.137202872	qg
Q3 (Atvirškktinis vertimas)	2.272461074	20.758246607	7.645236288	3.931846938	2.272461074	0.1477826229	0.04504184174	0.142505453	qg
Q4 (Parafrazavimas)	2.110366908	20.826077821	7.539717587	3.764856312	2.110366908	0.1455665249	0.0436360896	0.1391110833	qg
Q5 (Sinonimų keitimas)	2.092798618	20.501188431	7.250275306	3.65820813	2.092798618	0.1401299794	0.04054471581	0.1350217491	qg

2 priedas. Klausimų generavimo modelių išvesčių pavyzdžiai

Kontekstas	Modelis	Sugeneruotas klausimas	Tikras klausimas

„Yale Provost“ biuras paleido keletą moterų į garsias universiteto pirmininkus. 1977 m. Hanna Holborn Gray buvo paskirta pareigas einanti Jele prezidente iš šios pareigos ir tapo Čikagos universiteto prezidente, pirmoji moteris, kuri buvo visavertė didžiojo universiteto prezidentė. 1994 m. Yale Provostas Judith Rodin tapo pirmuoju Pensilvanijos universiteto „Ivy League“ institucijos prezidente. 2002 m. Provostas Alisonas Richardas tapo Kembridžo universiteto vicekancleriu. 2004 m. Provostas Susan Hockfield tapo Masačusetso technologijos instituto prezidente. 2007 m. Provosto pavaduotojas Kim Bottomly buvo paskirtas Wellesley koledžo prezidentu. 2003 m. Divincijos mokyklos dekanas Rebecca Chopp buvo paskirtas Colgate universiteto prezidentu ir dabar vadovauja Swarthmore koledžui.	Q1 (20k)	buvo paskirtas Harvardo universiteto prezidentu?	Kuriais metais Judith Rodin tapo pirmąja „Ivy League“ mokyklos prezidente?
„The Times“ naudojo reikšmingų veikėjų indėlį politikos, mokslo, literatūros ir menų srityse, kad sukurtų savo reputaciją. Didžiąją savo ankstyvojo gyvenimo dalį „The Times“ pelnas buvo labai didelis, o konkurencija buvo minimali, todėl jis gali mokėti daug geriau nei jos konkurentai už informaciją ar rašytojus. Nuo 1814 m. Straipsnis buvo atspausdintas ant naujojo garo varomo cilindro preso, kurį sukūrė Friedrichas Koenigas. 1815 m. „Times“ apyvarta buvo 5000.	Q2 (40k)	Koks buvo „Times“ pelnas?	Kas 1814 m. Sukūrė naują spaudos tipą?
Loginis garso kompaktinio disko (oficialiai kompaktinio disko skaitmeninio garso ar CD-DA) formatas aprašytas 1980 m. Dokumente, kurį sukūrė formato jungtiniai kūrėjai „Sony“ ir „Philips“. Dokumentas šnekama kaip „Red Book CD-DA“ po jo viršelio spalvos. Formatas yra dviejų kanalų 16 bitų PCM, koduojantis 44,1 kHz mėginių ėmimo greičiu viename kanale. Keturių kanalų garsas turėjo būti leistina „Red Book“ formatu, tačiau jis niekada nebuvo įgyvendintas. „Monoural Audio“ nėra esamo standarto „Red Book“ kompaktiniame diske; Taigi mono šaltinio medžiaga paprastai pateikiama kaip du identiški kanalai standartiniame „Red Book“ stereofoniniame takelyje (t. Y. Veidrodinis monoidas); Tačiau MP3 kompaktinis diskas gali turėti garso failų formatus su „Mono Sound“.	Q3 (Atvirkštinis vertimas)	Kokia yra „Red Book CD-DA“ formatas?	Koks yra oficialus kompaktinio disko pavadinimas?
Organizuotas nusikalstamumas jau seniai buvo siejamas su Niujorko miestu, pradedant keturiasdešimt vagių ir kuojos sargybinių penkiuose 1820 -aisiais taškuose. XX amžiuje pakilo mafija, kurioje dominavo penkios šeimos, taip pat gaujose, įskaitant juodus kastuvus. Mafijos buvimas mieste sumažėjo XXI amžiuje.	Q4 (Parafrazavimas)	Kokia buvo pagrindinė misija Niujorko miesto?	Kaip vadinosi pirmosios didelės nusikaltėlių gaujos Niujorke 1820-aisiais?
Franšizė pirmą kartą buvo sukurta 1997 m. Kaip virtualių augintinių serija, panaši į šiuolaikiškus „Tamagotchi“ ar „Nano Giga Pet“ žaislus, ir tai padarė įtaką stiliui. Būtybės pirmiausia buvo suprojektuoti taip, kad atrodytų mieli ir ikoniški net ir mažuose įrenginių ekranuose; Vėlesni pokyčiai juos sukūrė su sunkiau kraštutiniu stiliumi, kurį paveikė Amerikos komiksai. Franšizė įgavo pagreitį dėl savo pirmojo anime įsikūnijimo, „Digimon Adventure“ ir ankstyvojo vaizdo žaidimo „Digimon World“, abu išleisti 1999 m. Keli anime ir filmų sezonai, kuriuose buvo rodomi, ir vaizdo žaidimų serija išsiplėtė į tokius žanrus kaip vaidmenų vaidinimas, lenktynės, kovos ir MMORPG. Taip pat buvo išleistos ir kitos žiniasklaidos formos.	Q5 (Sinonimų keitimas)	Koks buvo ankstyvojo vaizdo žaidimų serija?	Kas turėjo įtakos digimonų išvaizdai?

3 priedas. Klausimų atsakymo modelių išvesčių pavyzdžiai

Kontekstas	Klausimas	Modelis	Sugeneruotas atsakymas	Tikras atsakymas
„Yale Provost“ biuras paleido keletą moterų į garsias universiteto pirmininkus. 1977 m. Hanna Holborn Gray buvo paskirta pareigas einanti Jele prezidente iš šios pareigos ir tapo Čikagos universiteto prezidente, pirmoji moteris, kuri buvo visavertė didžiojo universiteto prezidentė. 1994 m. Yale Provostas Judith Rodin tapo pirmuoju Pensilvanijos universiteto „Ivy League“ institucijos prezidente. 2002 m. Provostas Alisonas Richardas tapo Kembridžo universiteto vicekancleriu. 2004 m. Provostas Susan Hockfield tapo Masačusetso technologijos instituto prezidente. 2007 m. Provosto pavaduotojas Kim Bottomly buvo paskirtas Wellesley koledžo prezidentu. 2003 m. Divincijos mokyklos dekanas Rebecca Chopp buvo paskirtas Colgate universiteto prezidentu ir dabar vadovauja Swarthmore koledžui.	Kuriais metais Judith Rodin tapo pirmąja „Ivy League“ mokyklos prezidente?	A1 (20k)	mokyklos prezidentas	1994 m
1967 m. Sukūrė pagrindą Heinz Kloss, Abstand ir Ausbau kalbas, apibūdinant kalbų bendruomenes, kuri, nors ir politiškai, ir (arba) kultūriškai, apima daugybę tarmių, kurios, nors ir glaudžiai susijusios genetiškai, gali būti skirtingi iki tarpusavio supratimo.	Kuriais metais buvo sukurta „Abstand“ ir „Ausbau Framework“?	A2 (40k)	ir „Ausbau Framework“	1967 m
Korėja buvo laikoma Japonijos imperijos dalimi kaip pramonės kolonija kartu su Taivanu, ir abu buvo Didžiojo Rytų Azijos auginimo dalis. 1937 m. Kolonijinis generalinis gubernatorius Jirō Minii Korėjai sakė 23,5 mln. USD. Korėjos kultūrinė asimiliacija, uždraudimas ir Korėjos kalbos, literatūros ir kultūros studijavimas, siekiant pakeisti privalomą naudojimą ir tyrinėti jų kolegas Japonijoje. Nuo 1939 m. Gyventojai turėjo naudoti japonų vardus pagal Sōshi-Kimei politiką. Korėjos nauda karinei pramonei prasidėjo 1939 m., Japonijos ar Japonijos darbo jėga - 2 milijonai korėjiečių.	Koks buvo generalinis generalinis gubernatorius, kuris įpareigojo Korėjos kultūrinę asimiliaciją?	A3 (Atvirkštinis vertimas)	23,5 mln. USD	Generolas Jirō Minami
1348 m. Ir 1349 m. Portugalijoje, kaip ir likusi Europa, buvo nuniokota Juodoji mirtis. 1373 m. Portugalija sudarė aljansą su Anglija, kuri yra ilgiausia aljansas pasaulyje. Šis aljansas tarnavo abiejų tautų interesams per visą istoriją ir daugelis jį laiko NATO pirmtaku. Laikui bėgant tai peržengė geopolitinį ir karinį bendradarbiavimą (abiejų tautų interesų apsauga Afrikoje, Amerikoje ir Azijoje prieš prancūzų, Ispanijos ir Nyderlandų konkurentus) ir palaikė tvirtus prekybos ir kultūrinius ryšius tarp dviejų senų Europos sąjungininkų. Ypač Oporto regione iki šiol yra matoma anglų kalbos įtaka.	Kuriame Portugiško regione vis dar matoma anglų kalbos įtaka?	A4 (Parafrazavimas)	Europos sąjungininkų	Oporto regionas

Nors XXI amžiuje klasikinė muzika prarado didelę dalį savo tradicijos muzikinei improvizacijai, nuo baroko eros iki romantiškosios eros yra atlikėjų, kurie galėtų improvizuoti savo eros stiliumi, pavyzdžių. Baroko eroje vargonininkai improvizuodavo preliudus, klavišininkai, grojantys klavesinu, improvizuodavo akordus iš figurinių boso simbolių, esančių po boso kontinuo partijos boso natomis, ir tiek vokaliniai, tiek instrumentiniai atlikėjai improvizuodavo muzikines puošmenas. J.S. Bachas buvo ypač žymus dėl savo sudėtingų improvizacijų. Klasikinės eros metu kompozitorius-atlikėjas Mozartas buvo žymus dėl savo sugebėjimo improvizuoti melodijas skirtingais stiliais. Klasikinės eros metu kai kurie virtuozo solistai improvizuodavo koncerto dalis „Kadencija“. Romantiškos eros metu Bethovenas improvizuodavo fortepijonu. Norėdami gauti daugiau informacijos, skaitykite improvizaciją.	Kas improvizuotų preliudų baroko eroje?	A5 (Sinonimų keitimas)	klavišininkai, grojantys klavišininkai	vargonų atlikėjai