VILNIAUS UNIVERSITETAS

MATEMATIKOS IR INFORMATIKOS FAKULTETAS

PROGRAMŲ SISTEMŲ STUDIJŲ PROGRAMA

Kursinis darbas

Edukacinio turinio kūrimas naudojant dirbtinį intelektą

Developing educational content using artificial intelligence

Balys Žalneravičius

Darbo vadovas : Dr. Irus Grinis

Vilnius
2026

Padėka

Darbo autorius nuoširdžiai dėkoja darbo vadovui už kantrybę ir rekordiškai greitą grįžtamąjį ryšį bei jaunimo klubui „Liepkiemis“ už šiltą darbo vietą ir neišsenkančią kavą.

Santrauka

Darbe tiriamos didžiųjų kalbos modelių (angl. *Large Language Models*) panaudojimo galimybės edukacinio turinio kūrimui. Tyrimo metu buvo sukurta sistema, galinti generuoti kontrolinius klausimus iš keturių Naujojo Testamento Evangelijų naudojant „Gemini-2.5-flash“, „Mistral-medium“ ir „Mistral-small“ modelius. Eksperimento metu įgyvendintas automatinis kryžminio vertinimo (angl. *cross-evaluation*) procesas, kurio metu modeliai vertino vieni kitų sugeneruotus klausimus pagal šešiabalę kokybės skalę. Rezultatai parodė, jog tinkamiausiai užduotį atliko „Gemini-2.5-flash“ modelis, pasiekęs 99,7 % patikimumą, tačiau į šią statistiką reikia žiūrėti kritiškai. Iš viso buvo sugeneruoti ir sėkmingai įvertinti 3265 klausimai, kuriuos galima pasitelkti ugdymo procese. Darbe pateikiama detali sistemos architektūra.

Raktiniai žodžiai: dirbtinis intelektas (DI), didieji kalbos modeliai (LLM), automatinis edukacinio turinio generavimas, kryžminis vertinimas, klausimų ir atsakymų sistemos, ugdymo procesas, švietimo sistema.

Summary

Paper explores the application possibilities of Large Language Models (LLMs) for creating educational content. During the study, a system was developed capable of generating test questions from the four New Testament Gospels using ”Gemini-2.5-flash”, ”Mistral-medium”, and ”Mistral-small” models. An automated cross-evaluation process was implemented during the experiment, where models evaluated each other’s generated questions based on a 6-point quality scale. The results showed that the ”Gemini-2.5-flash” model performed the task most effectively, achieving 99.7% reliability. In total, 3,265 questions were generated and successfully evaluated. These questions are suitable for the educational process. The work provides a detailed architecture of the system.

Keywords: artificial intelligence (AI), large language models (LLMs), educational content generation, cross-evaluation, question and answer systems, educational process, education system.

Iliustracijų sąrašas

Lentelių sąrašas

Turinys

Įvadas

Švietimo sistemos iššūkiai yra viena esminių Lietuvos problemų šiuo metu. Švietimo, mokslo ir sporto ministerijos kartu su Nacionaline švietimo agentūra apžvalgoje išskiriami trys esminiai ugdymo proceso iššūkiai:

- **Mokytojų senėjimas:** 2024 m. duomenimis, net 61,1 % visų bendrojo ugdymo mokyklų mokytojų buvo 55 metų ir vyresni (2022 m. jų buvo 59,1 %).
- **Pasiekimų atotrūkis tarp kaimų ir miestų:** tris ir daugiau valstybinių brandos egzaminų (VBE) miestuose išlaiko 75,8 % abiturientų, o mažosiose savivaldybėse - 68,2 %.
- **Tėvų išsilavinimo įtaka:** asmenys, kurių tėvai turi aukštąjį išsilavinimą, patys jį įgyja 66 % atvejų, o tie, kurių tėvai nebaigė vidurinės mokyklos - tik 4 %. Atotrūkis vienas didžiausių tarp EBPO šalių.

Šie iššūkiai verčia ieškoti netradicinių sprendimo būdų, kurie mažintų galimybių atskirtį ir lengvintų mokytojų darbą. Šių švietimo iššūkių sprendimas gali būti generatyvinio dirbtinio intelekto (DI) sistemų integracija į ugdymo procesą.

Pastorojo penkmečio proveržis dirbtinio intelekto srityje tai didieji kalbos modeliai (angl. *Large Language Models*, toliau - LLM). *„ChatGPT-3.5“* kalbos modelis po viešo paleidimo 2022 metų lapkričio 30 dieną eksponentiniu greičiu paplito po pasaulį. Per penkias pirmąsias dienas *„OpenAI“* platforma pasiekė daugiau nei milijoną registruotų vartotojų. Po vienerių metų *„ChatGPT“* pasiekė 100 mln. aktyvių vartotojų, po dvejų metų - beveik 350 mln., o 2025 m. liepos pabaigoje jų skaičius viršijo 700 mln. Tai sudaro apie 10 % visos pasaulio suaugusiųjų populiacijos [CCD+25].

Sydnejaus technologijų universiteto mokslininkai analizavo 88 empirinius tyrimus, publikuotus nuo *„ChatGPT-3.5“* pasirodymo iki 2025 m. kovo [SYD+26]. Tyrėjai daro išvadą, jog LLM integravimas į ugdymo procesą teigiamai veikia akademinius pasiekimus, didina mokinių motyvaciją bei įsitraukimą ir leidžia efektyviau naudoti mokymo resursus. Visgi, aprašomi ir kritiniai iššūkiai: modelių haliucinacijos, moksleivių perteklinis pasitikėjimas DI įrankiais, vertinimo patikimumas bei duomenų privatumo ir saugumo užtikrinimas.

Generatyvinis dirbtinis intelektas sudaro galimybę mokymosi turinį individualizuoti pagal mokinio poreikius, tačiau tiesioginis didžiųjų kalbos modelių (pavyzdžiui *„ChatGPT“*) taikymas išlieka rizikingas dėl haliucinacijų pavojaus. Tai ypač aktualu dirbant su istoriniais tekstais ir kitais griežto faktinio tikslumo reikalaujančiai šaltiniais.

Darbo tikslas - sukurti automatizuotą testinių klausimų generavimo sistemą, naudojant keturių Evangelijų tekstus, ir empiriškai įvertinti jų kokybę pasitelkiant kryžminį vertinimą.

Siekiant šio tikslo, tyrime bus atlikta:

- Šaltinių apie edukacinio turinio kūrimą naudojant dirbtinį intelektą analizė.
- Parengta programa keturių Evangelijų tekstų išgavimui ir struktūrizuoto tekstų rinkinio sudarymui.

- Ekspermentinėje dalyje bus įgyvendintas automatinis klausimų generavimo procesas pasitelkiant tris LLM modelius.
- Automatinis sugeneruotų klausimų kryžminio vertinimo mechanizmas, klausimų kokybei įvertinti.
- Pabaigoje bus atrinkti aukščiausią įvertinimą gavę klausimai ir sudarytas katechezės ugdymo procesui tinkamų klausimų rinkinys.
- Klausimų generavimo, įvertinimo ir atrinkimo sistema galės bus naudojama su kitais šaltiniais.

Darbe aprašomas kiekybinis ekspermentinis tyrimas, siekiant įvertinti didžiųjų kalbos modelių (LLM) gebėjimą generuoti edukacinį turinį. Tyrimo duomenų šaltinis - keturi, Mato, Morkaus, Luko ir Jono Evangelijų tekstai. Pagrindinis tyrimo metodas - automatizuotas turinio generavimas pasitelkiant tris skirtingus LLM modelius (*Gemini-2.5-flash*, *mistral-medium-2508* ir *mistral-small-2506*). Programinis kodas tyrimui rašomas *Python* programavimo kalba, o universaliai API sąsajai užriktinti naudojama *Python LiteLLM* biblioteka [Lit]. Klausimų teisingimui įvertinti naudojamas vertinimo metodas (angl. cross-evaluation), kurio metu modeliai pagal aprašytą skalę įvertina klausimus. Galutinių rezultatų patikimumas nustatomas atliekiant ekspertinį vertinimą.

Darbą sudaro įvadinė dalis apie temos aktualumą, teorinė dalis apie DI istoriją ir taikymą ugdymo procese bei ekspermentinė dalis, kurioje aprašoma įgyvendinta klausimų generavimo ir vertinimo sistema bei išvados.

1. DI taikymai edukacinio turinio kūrimui

1.1. Raktiniai DI istorijos momentai

Šiandieninio dirbtinio intelekto šaknys yra neatsiejamos nuo 1950 m. daktaro Alano Turingo žurnale *„Mind“* paskelbto straipsnio „Computing Machinery and Intelligence“. Jame autorius iškėlė esminį klausimą: „Ar mašinos gali mąstyti?“ ir pasiūlė apie jį galvoti per „imitacijos žaidimą“. Jei mašina sugeba įmituoti žmogaus elgesį ir kalbą, tai reiškia ji intelektuali [Tur50].

Praėjus penkeriems metams po šios publikacijos, Turingo iškelta teorinė diskusija įgavo kūną. Vieną vasarą į Dartmuto koledžą, esantį Hanoveryje, Naujajame Hampšyre JAV, buvo pakviesti dešimt kompetentingų mokslininkų dviejų mėnesių trukmės bendroms studijoms. Pamatinė tyrimo hipotezė teigė, jog kiekvienas mokymosi aspektas ar intelekto savybė iš principo gali būti taip detaliai apibūdinta, kad mašina gebėtų ją simuliuoti. Šis kvetimas į vasaros studiją buvo paskelbtas 1955 metais ir yra laikomas istoriniu momentu, kai buvo oficialiai suformuluotas ir pasiūlytas terminas „dirbtinis intelektas“ (angl. *artificial intelligence*) bei numatyti pagrindiniai jo tyrimo aspektai: mašinos gebėjimas savarankiškai mokytis, naudoti kalbą bei kurti naujas žinias. [MMR+55].

Pirmasis DI kalbos modelis atsirado 1966 metais. Džozefas Veicenbaumas Masačusetso technologijos institute (MIT) sukūrė programą ELIZA. Ji veikė veidrodžio principu, kitais žodžiais tariant, prefrazuodavo žmogaus parašytą žinutę, taip sukurdama įvaizdį, jog supranta turinį. Nors iš tiesų tai buvo tiesiog algoritmas, kuris žmogaus užklausoje surasdavo raktinius žodžius lygindamas juos su iš anksto įrašytais žodžiais ELIZA atmintyje. Atrinkęs raktinį žodį, jį įstatydavo į sakinio šabloną, kurį grąžindavo žmogui. Dažniausiai tai būdavo klausimas. Pavyzdžiui: jei vartotojas parašo „Man sunku rašyti kursinį darbą.“. ELIZA aptinka raktinius žodžius „kursinis darbas“ ir gali atsakyti „Kodėl tau sunku rašyti kursinį darbą?“ [Wei66]. ELIZA dažnai vadinama „simboliniu“ DI (angl. *Symbolic AI*), nes ji dirba su simboliais (žodžiais) pagal programuotojo sukurtus šablonus.

Plėtojantis simboliniam dirbtiniam intelektui, aštuntajame XX a. dešimtmetyje iškilo taisyklėmis grįstos sistemos (angl. *Rule-based AI*), dar žinomos kaip ekspertinės sistemos. Kitaip nei *ELIZA*, šios sistemos turėjo sukauptas milžiniškas žinių bazes, sudarytas iš šimtų „jeigu-tada“ (angl. *if-then*) taisyklių. Jos veikė kaip srities ekspertai: priklausomai nuo įvestų duomenų, sistema logiškai susiedavo taisykles ir pateikdavo pagrįstą išvadą. Viena reikšmingiausių to meto programų - MYCIN, skirta kraujo infekcijoms diagnozuoti [Sho77].

Tačiau simbolinis DI susidūrė su problema - taisyklėmis buvo neįmanoma aprašyti visų galimų atvejų. XX a. paskutiniame dešimtmetyje įvyko lūžis, ypač reikšmingas DI taikymo edukacijai - mašinis mokymasis (angl. *Machine Learning*) tapo dominuojančia paradigma. Užuot rėmęsi griežtomis, žmogaus įrašytomis taisyklėmis, mašininio mokymosi algoritmai naudojo statistinius metodus, leidžiančius pačioms sistemoms atrasti ir išmokti dėsningumus iš pateiktų tūkstančių pavyzdžių. [RN10]

Kitas šuolis įvyko giliojo mokymosi (angl. *Deep Learning*) dėka. DI perėjo prie daugiasluoksnių neuroninių tinklų architektūrų, gebančių išmokti dar sudėtingesnius desningumus iš duomenų rinkinių. 2012 m. *ImageNet* konkurse gilieji konvoliuciniai tinklai beveik perpus sumažino klaidų lygį lyginant su kitais metodais [LBH15]. Tai sudarė sąlygas šiuolaikinių generatyvinių kalbos modelių atsiradimui, kurie ir naudojami edukaciniam turiniui kurti.

Generatyvinio dirbtinio intelekto (angl. *Generative AI*) sistemos perėjo nuo duomenų klasifikavimo prie naujo turinio kūrimo. Šis šuolis įvyko panaudojant giliojo mokymosi paradigma su milžiniškais tekstinių duomenų rinkiniais, leidžiančiais modeliams išmokti sudėtingas kalbines struktūras. Skirtingai nuo ankstesnių sistemų, kurios rėmėsi statiškomis taisyklėmis, generatyviniai modeliai, tokie kaip *ChatGPT, Gemini ar Mistral*, yra apmokyti numatyti labiausiai tikėtiną sekos žodį pagal pateiktą kontekstą. Tai suteikia kalbos modeliams gebėjimą ne tik interpretuoti vartotojo užklausas, bet ir generuoti rišlius, kontekstą atitinkančius ir edukacijai tinkamus tekstus. [CCD+25].

1.2. DI panaudojimo galimybės ugdymo procese

Sydnejaus technologijų universiteto mokslininkai 2025 m. publikavo straipnį - *„Didieji kalbos modeliai švietime: sisteminė empirinių taikymų, privalumų ir iššūkių apžvalga“*. Straipsnio autoriai detaliai išanalizavo 88 empirinius tyrimus LLM taikymo edukacijai tema, publikuotus nuo *„ChatGPT-3.5“* pasirodymo 2022 m. vasario 20 iki 2025 m. kovo. Šioje apžvalgoje susisteminami naujausi empiriniai tyrimai, apimantys šešias funkcines LLM taikymo edukacijai kategorijas: pokalbių robotus, mokomojo turinio generavimą, automatizuotą vertinimą ir grįžtamąjį ryšį, palaikymo įrankius, ugdymo proceso priemones bei intelektualiąsias mokymo sistemas. Kartu pateikiamos įrodymais pagrįstos įžvalgos apie šių technologijų diegimą į ugdymo procesą.

LLM pagrindu sukurti **pokalbių robotai** sudaro galimybę besimokančiajam turėti betarpiškai prieinamą, asmeninį asistentą. Tai leidžia mokiniui dirbti individualiu tempu bei gauti pagalbą realiu laiku. Kita esminė pokalbių robotų savybę - pokalbių tęstinumas. Kitaip nei pavieniai klausimai mokytojui, LLM atsakydama atsižvelgia į prieš tai vykusių pokalbių kontekstą.

Mokomojo turinio generavimas naudojant LLM sprendžia dideles laiko sąnaudas bei priklausomybę nuo mokytojo kompetencijos. LMM leidžia generuoti interaktyvų mokomąjį turinį, kuris gali būti personalizuojamas pagal besimokančiojo amžių ar individualius poreikius. LLM gebėjimas pritaikyti turinį didina prieinamumą bei naikina kalbinius barjerus. Tyrimai rodo, kad tinkamai suformuluotos LLM užklausos leidžia sukurti turinį, savo kokybe prilygstantį specialistų parengtai medžiagai. Itin svarbu tai, kad DI sugeneruotos užduotys, prižiūrimos mokytojų, gali netgi pranokti tradiciniais metodais parengtus kontrolinius darbus.

Automatizuotas vertinimas ir grįžtamasis ryšys pasitelkiant LLM mažina mokytojų darbo krūvį ir sukuria sąlygas personalizuotai pagalbai didelėse besimokančiųjų grupėse. Tyrimai rodo, jog LLM sistemos geba pasiekti vertinimo patikimumą, kuris yra artimas žmogaus-eksperto vertinimui. Apart įvertinimo, LLM sistemos suteikia gilų, asmeninį grįžtamąjį ryšį, kuris skatina mokinio motyvaciją ir savarankiškumą. Visgi, DI negali pakeisti žmogiško mokytojo ir mokinio santykio, o vertinimo ir grįžtamojo ryšio kokybė tiesiogiai priklauso nuo užduoties sudėtingumo. Todėl ugdymo proceso rezultatai priklauso ne tik nuo LLM sistemos, bet ir nuo besimokančiojo gebėjimo kritiškai vertinti gautus rezultatus.

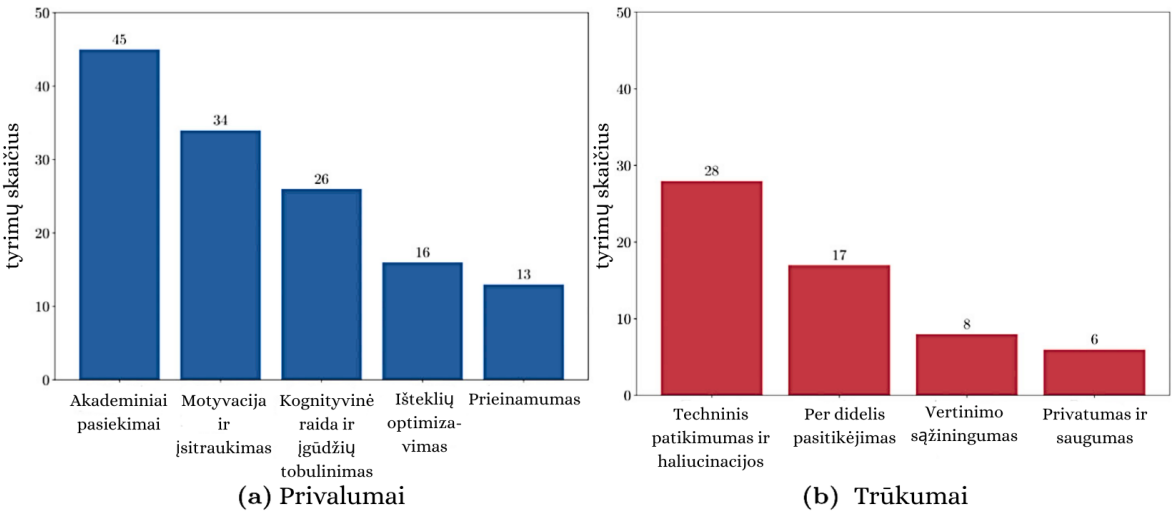
Savarankiško darbo palaikymo įrankiai LLM pagrindu veikia kaip pastoliai (angl. scaffolding), jie teikia nuolatinę pagalbą: užuominas, patarimus, atlieka gramatikos taisymą, kartu išlaikant autoriaus originalų stilių. Nors šie įrankiai veiksmingi, bet per didelis pasitikėjimas jais gali apriboti žinių įsisavinimą.

Ugdymo proceso priemonės LLM pagrindu keičia mokinių įsitraukimą į ugdymo procesą. Šie įrankiai įgalina interaktyvius pedagoginius metodus, tokius kaip „mokymasis mokant“ - besimokantysis turi pats paaiškinti medžiagą tarsi būtų mokytojas. Be to, tokios DI priemonės teikia mokytojams analitinius duomenis apie mokinio progresą. Tačiau tyrimai įspėja, jog sumažėjęs darbo krūvis gali lemti menkesnes mokinio pastangas.

Intelektualiosios mokymo sistemos (angl. Intelligent Tutoring Systems) yra viena perspektyviausių sričių. Šios sistemos pasižymi gebėjimu realiu laiku koreguoti turinį atsižvelgiant į mokinio žinias bei emocinius elgsenos faktorius (pavyzdžiui, nerimą ar dėmesio sutrikimą). Korekcijos atliekamos remiantis: užduočių atlikimo laikais, pagalbos prašymų dažnumu, klaidų kiekiu, skaitymo atidumu, gali būti sekami ir akių judesiai.

Tyrimas atskleidžia, kad **LLM taikymo ugdyme privalumai** apima geresnius akademinius rezultatus (n = 45), išaugusią studentų motyvaciją bei įsitraukimą (n = 34) ir spartesnį kognityvinių įgūdžių vystymąsi (n = 26). Taip pat pabrėžiamas resursų optimizavimas bei didesnis mokymosi medžiagos prieinamumas, leidžiantis gauti personalizuotą grįžtamąjį ryšį realiu laiku.

Visgi, **LLM taikymas ugdymo procese turi ir trūkumų**. Dažniausiai įvardijama problema - nepatikimumas ir haliucinacijos (n = 28). Modeliai generuoja faktines klaidas. Pedagoginiu požiūriu svarbiausia problema, tai per didelis pasitikėjimas DI įrankiais (n = 17), o tai mažina moksleivių kritinį mąstymą. Kito tyrime išskiriamos problemos: vertinimo sąžiningumas, algoritmų šališkumas bei duomenų privatumas ir saugumas. Žemiau pateiktoje 1 lentelėje iliutruojami LLM integravimo į ugdymo procesą privalumai bei trūkumai.



(a) Privalumai (b) Trūkumai

1 pav. *LLM taikymo ugdyme privalumų ir trūkumų pasiskirstymas (pagal Shi ir kt., 2026)*

Apibendrinant, taikant LLM ugdymo procese, tikrai gerėja akademiniai rezultatai bei kyla motyvacija, tačiau didžiausia problema išlieka techninis nepatikimumas (haliucinacijos) bei per didelė mokinio priklausomybė nuo įrankio, kuri silpnina kritinį mąstymą.

1.3. LLM taikymai didelių tekstų analizei edukacinio turinio generavimo tikslais

Istoriškai edukacinio turinio generavimas rėmėsi taisyklėmis grįstais metodais, jie naudojo rankiniu būdu sukurtas lingvistines taisykles (angl. *Part-of-Speech*). Nors šie metodai galėjo generuoti sintaksiškai teisingus klausimus, jie reikalavo didelių laiko sąnaudų bei specialisto kompetencijos, to pasekoje ribojo sistemos tinkamumą plataus mąsto edukacinio turinio generavimui [ARA24].

LLM panaudojimas šį laiko ir specialisto poreikį sprendžia. Edukacinio turinio kūrimas naudojant LLM remiasi keliomis pagrindinėmis metodikomis, kurios yra būtinos siekiant sukurti tinkamą mokymosi medžiagą. Vieno stipriausių Brazilijos universeto (Espirito Santo) mokslininkai 2024 m. atliko tyrimą, norėdami išsiaiškinti, kokios metodikos yra svarbiausios, siekiant automatiškai generuoti testinius (angl. *Multiple-Choice*) klausimus remiantis didžiaisiais kalbos modeliais. Šiame tyrime klausimų šaltinis buvo universiteto dokumentai (nuostatai, taisyklės, reglamentai) [MPM⁺24].

Užklausų inžinerija (angl. *Prompt Engineering*) Pagrindinis įrankis generuojant edukacinį turinį, o tiksliau klausimus yra užklausų formulavimas. Užklausoje kalbos modeliui priskiriamas specifinis vaidmuo **(angl. *Role-based prompting*)**, pavyzdžiui: „Esate profesorius, rengiantis klausimus studentams“, toks priskyrimas modelio atsakymus padaro aktualesnius ir padidina išvesties nuoseklumą. Klausimai sugeneruoti užklausoje nurodant vaidmenį surinko aukščiausius įvertinimus. [MPM⁺24]. Vieno pavyzdžio užklausa **(angl. *One-shot prompting*)**. Taikant šį metodą, kartu su įvestimi pateikiamas konkretus norimo rezultato formato ar stiliaus pavyzdys, padedantis modeliui suprasti ir sugeneruoti atsakymus, atitinkančius pateiktą pavyzdį. Šis būdas išnaudoja modelio gebėjimą apibendrinti informaciją iš minimalaus duomenų kiekio, todėl jis gali pateikti tikslius ir kontekstiškai tinkamus atsakymus remdamasis vos viena iliustratyvia užklausa. [MPM⁺24].

Rezultato savianalizė ir vertinimas Tai žingsnis, padedantis užtikrinti rezultato kokybę be nuolatinės žmogaus intervencijos. Brazilijos mokslininkų tyrime po pirminio klausimo sugeneravimo seka peržiūros etapas, kurio metu modelis tikrina savo paties darbą pagal šiuos kriterijus: tikrinama, ar išlaikytas JSON formatas ir ar tekstas neturi rašybos klaidų bei tikrinama, ar turi tik vieną teisingą atsakymo variantą. [MPM⁺24]. Tyrime teigiama, kad vertinimo sistema turėtų naudoti kitą LLM modelį nei generavimo sistema, siekiant išvengti šališkumo. Jei tas pats modelis naudojamas ir generavimui, ir vertinimui, jis yra linkęs atleisti savo paties padarytas klaidas arba jų tiesiog nepastebėti [MPM⁺24]. Indonezijos mokslininkai 2024 metais tyrė edukacinių klausimų vertinimą ir klasikavimą bei pastebėjo, jog modeliai žymiai geriau klasifikuoja ir vertina savo pačių sugeneruotus klausimus. Nustatyta, kad *„ChatGPT“* klasifikavimo tikslumas (F1 balas) buvo 0,76, kai jis vertino savo paties sukurtus klausimus, lyginant su vartotojų sugeneruotų klausimų vertinimu 0,64. Tai rodo modelio šališkumą savo paties klausimams. Taigi, nors savianalizė tinkama struktūros patikrinimui, ji praleidžia kokybės trūkumus, kurie būtų pastebėti vertinant nepriklausomu modeliu [ARA24].

Svarbiausios informacijos vieta užklausoje ir jos apimtis. Stanfordo ir Berklio Kalifornijos universitetų mokslininkai 2023 m. atliktame tyrime atskleidė, jog didžiųjų kalbos modelių efektyvumas priklauso nuo informacijos vietos kontekste: modeliai geriausiai supranta duomenis, esančius ilgo teksto pradžioje arba pabaigoje. Pavyzdžiui, kai *„GPT-3.5-Turbo“* turėjo rasti atsakymą dokumentų rinkinio viduryje, jo tikslumas krito daugiau nei 20 %, lyginant su atvejais, kai informacija buvo pradžioje. Tyrime tai pat atrasta, jog nors LLM modelis oficialiai palaiko didelį kiekį žetonų (angl. *tokens*),

pavyzdžiui, 16 000 ar 100 000, tai nereiškia, kad jis visą šį kontekstą naudoja vienodai efektyviai. Tyrėjai daro išvadą, kad efektyviau yra pateikti mažiau, bet kokybiškesnių šaltinių [LLH+23].

Taigi, siekiant sugeneruoti kokybiškiausius klausimus, svarbu: turinio šaltinį skaidyti dalimis, formuluojant užklausą modeliui suteikti vaidmenį, pateikti pageidaujamos klausimo formos pavyzdį. Sugeneravus klausimus, juos pateikti modeliams savianalizei įvertinti struktūrai, o klausimo turinio vertinimą, haliucinacijų aptikimą reikia patikėti kitam LLM modeliui, o ne tam, kuris klausimą sugeneravo.

2. Eksperimentinis klausimų generavimo ir vertinimo tyrimas

Šiame skyriuje aprašomas praktinis metodologijos pritaikymas ir turinio generavimo sistemos sukūrimas naudojant pasirinktus LLM modelius bei pabaigoje pateikiami gauti tyrimo rezultatai.

2.1. Klausimų generavimo ir vertinimo procesas

2 diagramoje pavaizduota trijų etapų sistema, skirta automatiškai generuoti kokybišką ir patikrintą edukacinį turinį naudojant LLM aplikacijų programavimo sąsają (angl. *Large Language Model Application Programming Interface*, toliau – LLM API). Kairėje diagramos dalyje matomos API užklausos, centre raktiniai etapai. Rodyklė aplink įvykį reiškia ciklą. Jei modeliai trys, tai subprocesas vykdomas tris kartus. Dešinėje diagramos pusėje pavaizduota duomenų išsaugojimo struktūra failų sistemoje. Tyrimo procesas susideda iš keturių pagrindinių etapų:

Duomenų paruošimo etapas (angl. *Data Preprocessing*): Procesas pradedamas nuo šaltinio tekstų (Evangelijų) išgavimo iš portalo *biblija.lt* ir išsaugojimo. Tekstas yra automatiškai nuskaitomas ir tvarkingai išsaugomas skyriais.

Generavimo etapas (angl. *Generation*): suformuluojama klausimų generavimo užklausa (angl. *prompt*), į kurią įterpus šaltinio tekstą, užklausa siunčiama vieno iš pasirinktų LLM modelių API sąsajai. Modelis, remdamasis *one-shot prompting* pavyzdžiu, sugeneruoja apskaičiuotą kiekį klausimų ir atsakymų JSON formatu.

Kryžminio vertinimo etapas (angl. *Cross-evaluation*): Vienas modelis sugeneruoja, o kiti du nepriklausomi modeliai vertina: jie lygina klausimo turinį su šaltinio tekstu, ieško loginių klaidų ar haliucinacijų, vertina didaktiką ir skiria įvertį 0–5 balų skalėje. Jei modeliai trys, tai kiekvienas įvertina kitų dviejų sukurtus klausimus.

Paskutinis etapas yra duomenų atrinkimas, tik aukščiausią (5) įvertinimą iš visų modelių gavę klausimai yra įtraukiami į galutinį klausimų rinkinį.

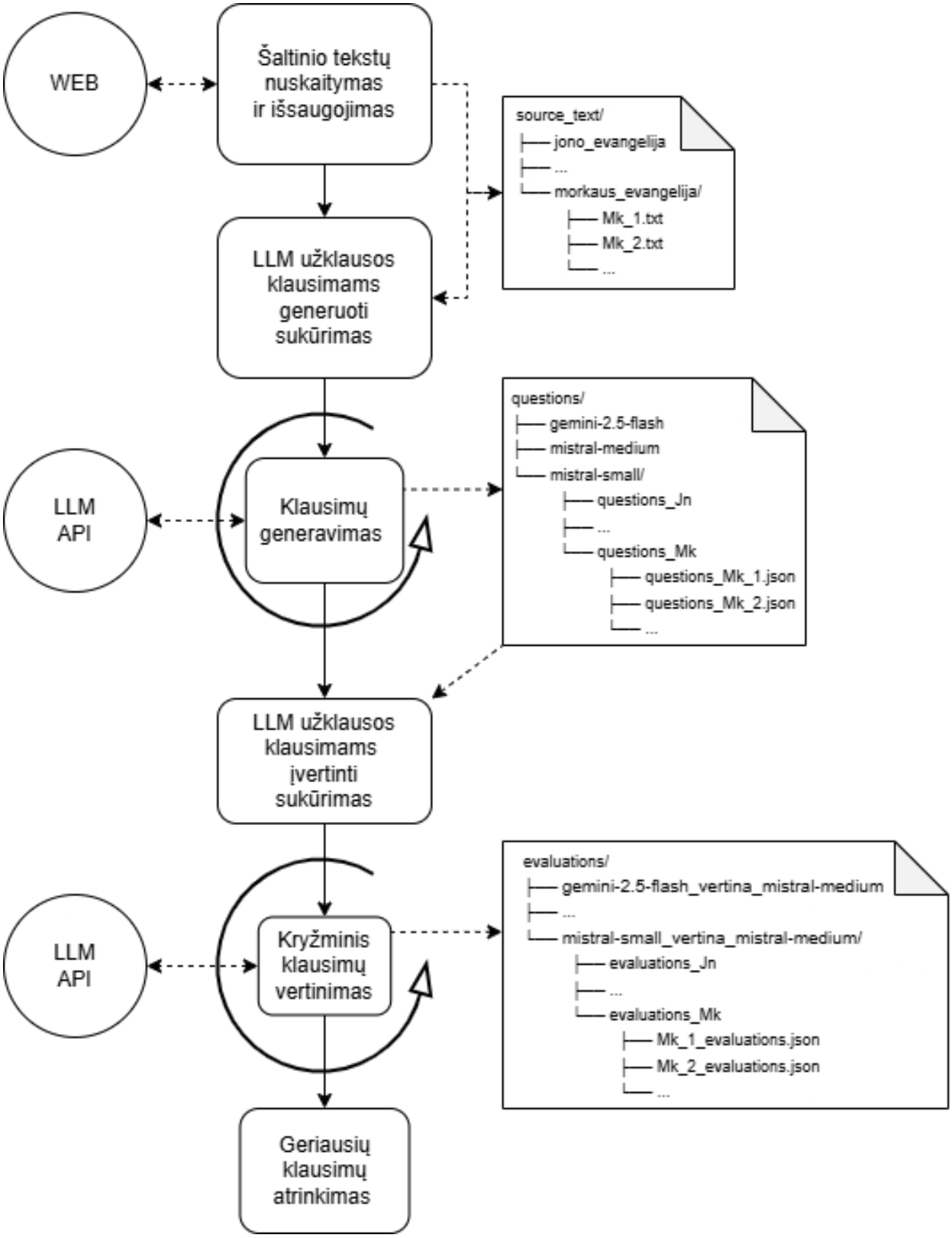
2.2. Šaltinio tekstų rinkinio sudarymas

Šaltinio tekstai, iš kurių bus generuojami klausimai, yra keturios Naujojo Testamento Evangelijos (Marko, Mato, Luko ir Jono). Naudotas lietuviškas kun. Česlovo Kavaliausko 1988 m. Naujojo Testamento vertimas iš graikų kalbos. Pasiekiamas skaitmeniniu formatu portale `biblija.lt` [dra24].

1 lentelė. *Informacija apie evangelijas [Cen24]*

Evangelija	Skyrių sk.	Istorinė parašymo data
Pagal Matą	28	45–59 m.
Pagal Morkų	16	42–45 m.
Pagal Luką	24	59 m.
Pagal Joną	21	60–70 m.

Šaltinio tekstai buvo išgauti automatizuotu būdu iš portalo *biblija.lt*. Šiam tikslui įgyvendinti



2 pav. *Automatinio klausimų generavimo ir vertinimo proceso diagrama*

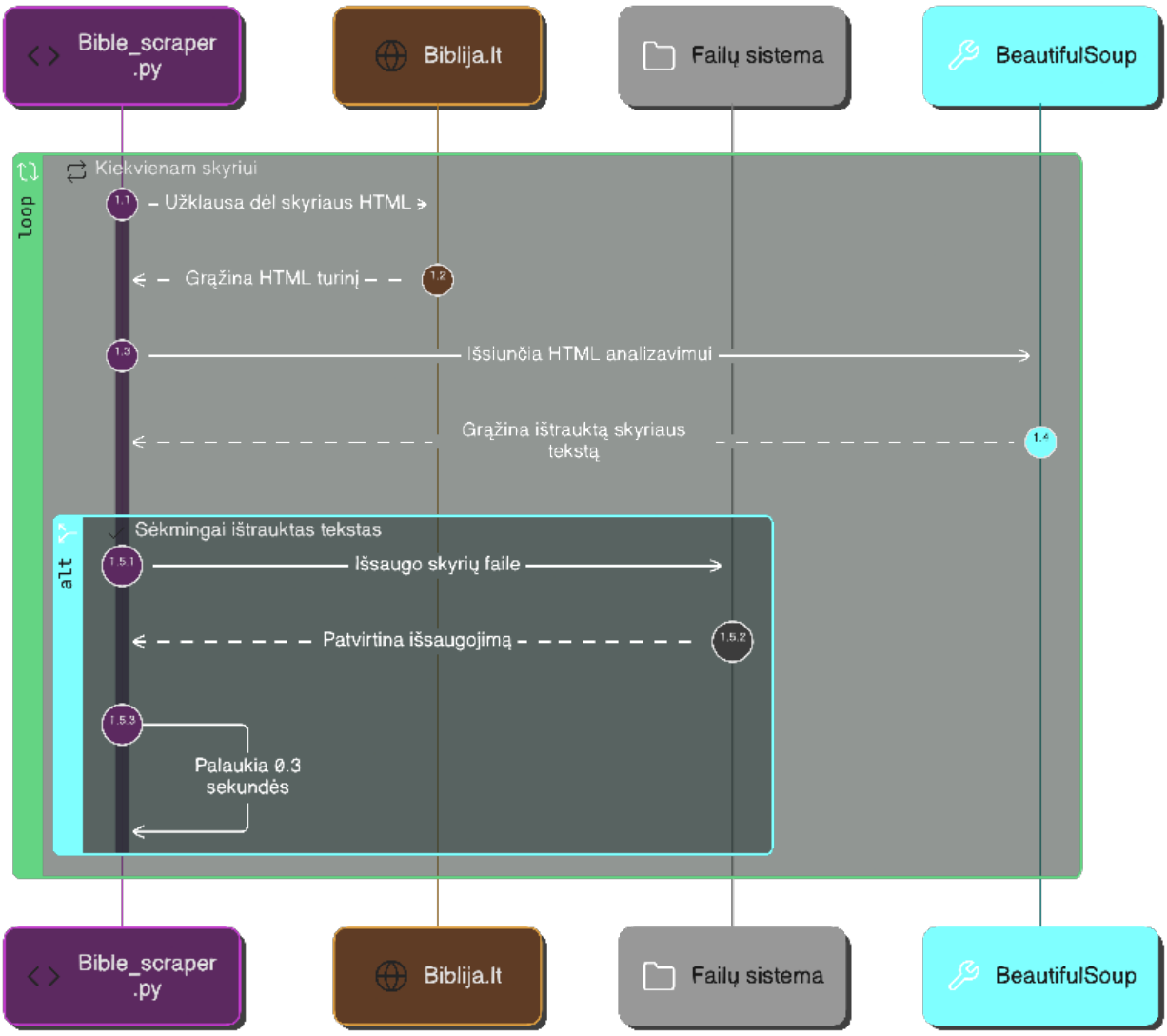
parašytas specializuotas duomenų gavybos (angl. *web scraping*) skriptas *Python* kalba, naudojant *BeautifulSoup4* biblioteką semantinei HTML turinio analizei [Ric25]. Tekstas automatiškai suskaidytas ir išsaugotas atskiruose tekstiniuose failuose (`.txt`). Vienas failas vienam skyriui. Iliustracijoje 3 vaizduojama, kaip kiekvieno tekstinio failo pradžioje yra nurodoma skyriaus santrumpa (Evangelijos autoriaus dvi vardo raidės ir skyriaus numeris).

```
Jn 1
1 Pradžioje buvo Žodis.
Tas Žodis buvo pas Dievą,
ir Žodis buvo Dievas.
...
```

3 pav. *Šaltinio tekstinio failo (`.txt`) fragmentas su Evangelijos skyriaus santrumpa*

Teksto skaidymą mažesnėmis apimtimis grindžia [LLH+23] tyrimas, siekiant neviršyti modelių efektyvaus konteksto lango. Tekstų išgavimo metu pašalintos perteklinės HTML žymos, tinklalapio navigacijos elementai bei išnašų nuorodos, tačiau išsaugoti eilučių numeriai. Visi skyriaus failai buvo patalpinti atskiruose aplankaluose pagal Evangelijų pavadinimus.

4 paveikslėlis yra sekos diagrama, joje detalizuojama, kaip buvo suformuotas tyrimo tekstinių šaltinių rinkinys. Procesas yra automatizuotas ir valdomas *Python* skripte, kuris veikia ciklišk ai, kol išgauna bei išsaugo visus suplanuotus Evangelijų skyrius failų sistemoje. Nors diagramoje tai nedetalizuojama, programiniame kode papildomai realizuotas kiekvieno žingsnio duomenų validavimas. Po sėkmingo portalo nuskaitymo programa atlieka priverstinę 0,3 sekundės pauzę. Tai apsaugo portalo serverį nuo perkrovimo ir užtikrina stabilų duomenų surinkimą.



4 pav. *Šaltinių duomenų rinkinio sudarymo eigos diagrama*

2.3. Modelių pasirinkimas

Kryžminiam patikrinimui reikalingi mažiausiai trys dalyviai, kad eliminuoti vieno modelio šališkumą, tad ir pasirinkti trys modeliai: ***Gemini-2.5-flash***, ***Mistral-medium-2508*** ir ***Mistral-small-2506***. Šie modeliai pasirinkti dėl jų prieinamumo. Tai buvo našūs bei aukštus rezultatus demontruojantys modeliai, tuo pačiu suteikiantys nemokamą ir aukštus limitus turinčią API sąsają [Gem25; RJL$^+$25].

2 lentelė. *Modelių lyginamoji lentelė [Gem25; RJL$^+$25]*

Modelis	Išleidimo data	Kontekstas	AIME ’24	LCB v5	GPQA
Gemini-2.5-Flash	2025 06	**1 mln.**	72 %	59,3 %	**82,8** %
Mistral-medium-2508	2025 08	128 tūkst.	**73,6** %	**59,4** %	70,8 %
Mistral-small-2506	2025 08	128 tūkst.	70,7 %	55,8 %	68,2 %

AIME’24 – matematika.
LiveCodeBench v5 (LCV v5) – programavimas.
GPQA – mokslinis mąstymas.

Lyginamojoje lentelėje matome, kad **Gemini-2.5-Flash** modelis išsiskiria savo konteksto dydžiu, net 1 mln. žetonų ir **GPQA** - mokslinio mąstymo rezultatas (82,8 %) pats aukščiausias, o tai svarbiausia analizuojant didelius tekstus ir formuluojant klausimus. Modelis **mistral-medium-2508** geriausiai pasirodė **AIME'24** (matematikos) (73,6), **LCV v5** (programavimo) (59,4) testuose, tačiau **mistral-small-2506** ir **Gemini-2.5-Flash** atsiliko nedaug [Gem25; RJL+25].

2.4. Tyrimo aplinka

Eksperimentui įgyvendinti pasirinkta *Python* (v3.14) programavimo kalba. Dažnai DI projektuose naudojama dėl plačios bibliotekų pasiūlos darbui su DI modeliais bei duomenų analizei. Komunikacijai su skirtingų gamintojų LLM modeliais pasitelkta *LiteLLM* (v1.81) biblioteka, kuri užtikrina standartizuotą užklausų formatą, vienodą sąveiką su *Gemini* ir *Mistral* API taškais [Lit]. HTML Tekstų apdorojimui naudota *BeautifulSoup4*, o rezultatų sisteminimui ir analizei – *Pandas* (v.2.2) biblioteka [Pyt25].

Siekiant užtikrinti eksperimento atkartojamumą, atviras programinis kodas, ištekliai ir sugeneruoti klausimai saugomi viešoje *GitHub* saugykloje. Projekto failų sistema suskirstyta hierarchiškai (žr. 5 pav.):

- **source_text/** – šaltinio Evangelijų tekstai, suskaidyti pagal autorius ir skyrius (`.txt` formatu).
- **results/questions/** – LLM modelių sugeneruoti klausimai, sugrupuoti ugrupuoti pagal modelio tipą ir Evan gelijos knygą.
- **results/evaluations/** – galutiniai kryžminio vertinimo rezultatai su balais ir tekstiniais komentarais.

Saugumo sumetimais API prieigos raktai buvo saugomi izoliuotame `.env` faile, kad duomenys nebūtų publikuojami kartu su programiniu kodu ir taip apsaugant nuo API resursų švaistymo.

Viso tyrimo vystymo metu buvo naudojama *Git* versijų kontrolės sistema. Tai leido užtikrinti nuoseklų programinio kodo pakeitimų sekimą ir apsaugojo nuo nelaimingų atsitikimų.

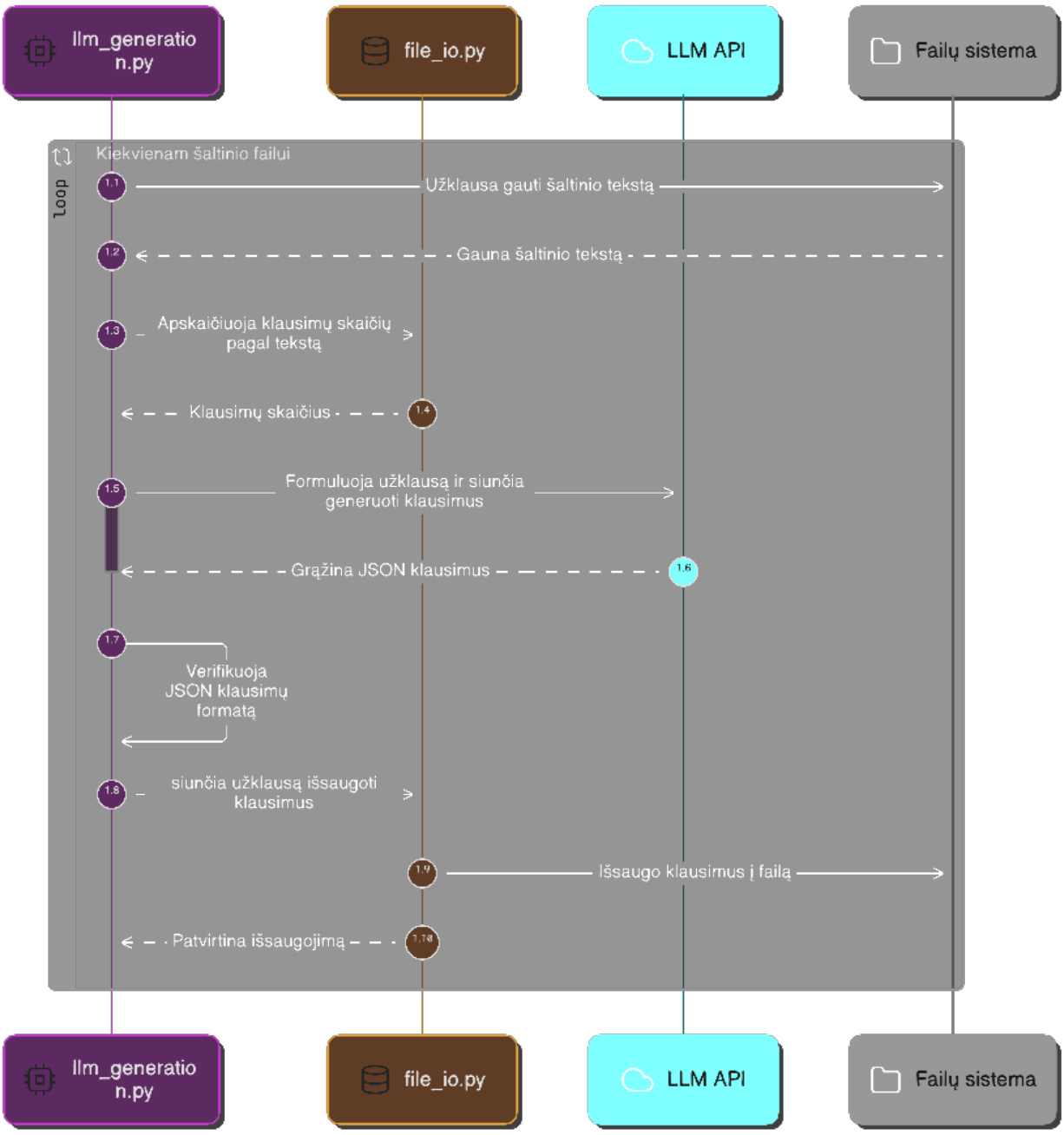
Visas tyrimo programinis kodas buvo rašomas ir testuojamas *Visual Studio Code* (v1.108) integruotoje kūrimo aplinkoje.

```
Project/
   bin/
   results/
       evaluations/
       questions/
           gemini-2.5-flash/
           mistral-medium/
           mistral-small/
               klausimai_Jn/
               klausimai_Lk/
               klausimai_Mk/
               klausimai_Mt/
                   questions_Mt_1.json
                   questions_Mt_2.json
                   ...
   source_text/
       jono_evangelija/
       luko_evangelija/
       mato_evangelija/
       morkaus_evangelija/
           Mk_1.txt
           Mk_2.txt
           ...
   src/
       main.py
       m...
   README.md
```

5 pav. *Projekto direktorijų struktūra*

2.5. Klausimų generavimas

Diagramoje 6 detalizuojama automatinio klausimų generavimo seka. Visos funkcijos, susiję su klausimų generavimu, yra aprašytos *llm_generation.py* programinio kodo modulyje. Generavimas vykdomas cikle: pradžioje nuskaitomas šaltinio skyriaus tekstas, pagal 1 formulę apskaičiuojamas reikiamas klausimų kiekis ir formuluojama užklausa, susidedanti iš sistemos bei vartotojo dalių (žr. 7 pav. ir 8 pav.). Tik tada užklausa siunčiama LLM modeliui per API sąsają. API grąžina klausimus JSON formatu (žr. 9 pav.). Klausimai verifikuojami, tikrinama, ar LLM grąžintas tekstas atitinka sintaksinius JSON reikalavimus ir ar nebuvo sugeneruota papildomų tekstinių simbolių už duomenų struktūros ribų. Taip pat, prieš išsaugant, pridedami klausimo *id*, *model* ir *chapter* laukai.



6 pav. *Automatinio klausimų generavimo eigos diagrama*

LLM užklausa susideda iš dviejų dalių: sistemos turinio, kuriame nurodomas vaidmuo modeliui (žr. 7 pav.), ir vartotojo turinio (žr. 8 pav.), jame aprašomas užklausos turinys, JSON klausimo pavyzdys [MPM+24]. Viena API užklausa skirta vieno skyriaus klausimų rinkiniui su keturiais atsakymų variantais sugeneruoti. Sistemos užklausa apibrėžia modelio veikimo ribas.

```
system_prompt = "Naudok taisyklingą lietuvių kalbą.
Niekada nepraleisk raidžių."
```

7 pav. *Sistemos užklausos dalis (angl. system prompt)*

```
user_prompt = "Sukurk " +number_of_questions+ "
klausimus su keturiais atsakymų variantais (a, b,
c, d) iš pateikto Biblijos teksto. "
"Tik vienas atsakymas turi būti teisingas. "
"Grąžink atsakymą IŠSKIRTINAI JSON formatu. JSON
struktūra turi būti tokia:"
"{"
'  "questions": ['
"    {"
'      "question_text": "...",'
'      "options": {"a": "...", "b": "...", "c": "...", "d": "..."},'
'      "correct_answer": "a"'
"    },"
"    {... (antras klausimas) ...},"
"    {... (trečias klausimas) ...}"
"  ]"
"}"
"Neįtraukite jokių paaiškinimų ar papildomo teksto,
tik JSON." + bible_text...
```

8 pav. *Vartotojo užklausos dalis (angl. user prompt)*

1 formulė skirta apskaičiuoti klausimų skaičių. Jis randamas skyriaus eilučių kiekį padalinus iš trijų. Taip pasirinkta siekiant, kad kiekvienam klausimui vidutiniškai tektų trys teksto eilutės ir būtų išlaikytas optimalus klausimų kiekis nepriklausomai nuo skyriaus dydžio.

$$KlausimuKiekis = \max\left(1, \frac{EiluciuSkyriujeKiekis}{3}\right) \quad (1)$$

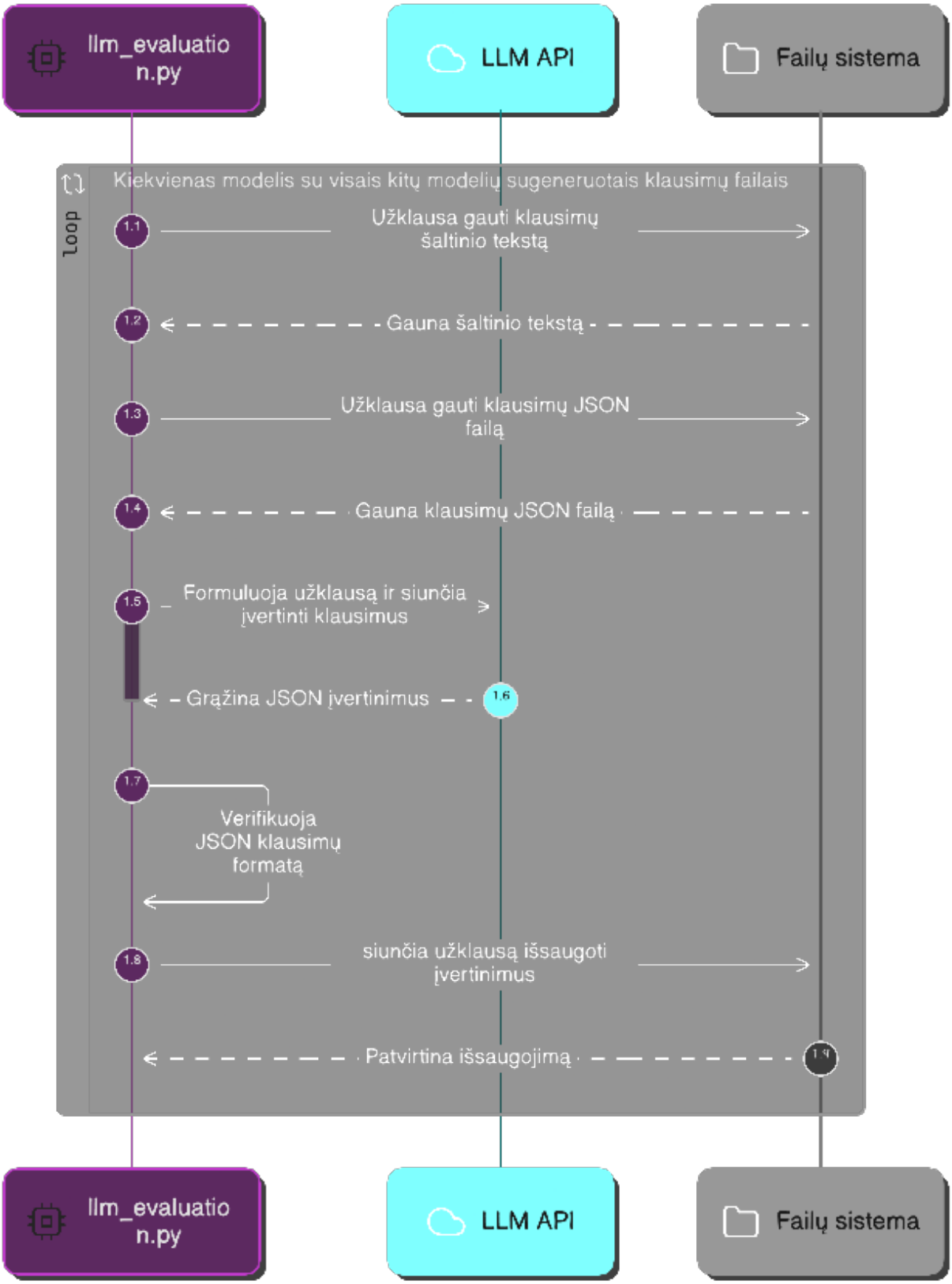
8 paveikslėlyje, *bible_text* vietoje įterpiamas skyriaus, iš kurio generuojami klausimai, tekstas. Užklausos formulavimas ir klausimų generavimas yra cikle, kuris kartojamas tol, kol pasibaigia skyrių `.txt` failai. Sugeneruoti klausimai validuojami ir išsaugomi JSON formatu. Išsaugoto klausimo pavyzdys randamas 9 paveikslėlyje.

```
{
    "id": "Jn_1_001",
    "question": "Kas buvo pradžioje, pasak Jono evangelijos 1 skyriaus?",
    "options": {
      "a": "Pasaulis",
      "b": "Žodis",
      "c": "Šviesa",
      "d": "Gyvybė"
    },
    "correct_answer": "b",
    "model": "gemini/gemini-2.5-flash",
    "chapter": "Jn_1"
}
```


9 pav. *Išsaugoto JSON klausimo pavyzdys*

2.6. Kryžminis sugeneruotų klausimų vertinimas

10 sekos diagrama detalizuoja tyrimo etapą, kuriame trys nepriklausomi LLM modeliai atlieka vienas kito sugeneruoto turinio vertinimą. Šis kryžminio vertinimo (angl. *cross-evaluation*) metodas pasirinktas siekiant minimizuoti pavienio modelio šališkumą bei užtikrinti aukštą generuojamų klausimų faktinį tikslumą.



10 pav. *Automatinio klausimų vertinimo eigos diagrama*

Sugeneravus klausimus su visais trimis LLM modeliais, kiekvienas modelis kryžmiškai įvertina kitų dviejų modelių sukurtus klausimus. Sistema aptinka, jei modelis bandytų vertinti savo klausimus,

0 – Visiškai netinkamas (neaiškus).

1 – Aiškus, bet faktiškai klaidingas (prieštarauja šaltiniui).

2 – Faktiškai teisingas, bet turi didelių turinio trūkumų (neteisingi atsakymai, klaidinanti logika).

3 – Teisingas, bet yra techninių/formos klaidų (gramatika, citavimo tikslumas).

4 – Puikus turinys ir technika, bet stilius/formuluotė galėtų būti geresni.

5 – Idealus visais aspektais (turinys, logika, gramatika, didaktinė vertė).

11 pav. *Klausimų vertinimo skalė LLM užklausai*

```
system_prompt =  "Tu esi Šv. Rašto ekspertas. Tavo užduotis
- įvertinti klausimų ir atsakymų kokybę (grade)."
"Vertinimo skalė:" vertinimo_skalė

"Vertink griežtai hierarchiškai: jei klausimas faktiškai
neteisingas, jis negali gauti daugiau nei 1 balo,"
"net jei jo gramatika ideali. Jei klausimas teisingas, bet
neaiškus, jis negali gauti daugiau nei 4 balų."

"Atsakymą pateik tik JSON formatu kaip sąrašą objektų,
atitinkančių šią struktūrą:"
"["
"  {"
"    "id": "klausimo_id","
"    "grade": įvertinimas"
"    "comment": 1-2 sakinių vertinimo paaiškinimas"
"  }"
"]"
```

12 pav. *Klausimų vertinimo sistemos užklausos dalis*

kad užtikrinti vertinimo objektyvumą. Klausimai vertinami pagal klausimų vertinimo skalę (žr. 11). Šešiabalė skalė leidžia vertinti ne tik faktinį teisingumą, bet ir didaktinę ir rašybos kokybę.

Klausimų vertinimui formuluojama LLM API užklausa. Joje įrašomas vaidmuo, laukiamo rezultato pavyzdys, šaltinio ištrauka, iš kurio klausimai buvo generuoti bei visi vieno modelio klausimai iš to skyriaus. Užklausa grąžina ne tik kiekvieno klausimo įvertinimą, bet ir trumpą komentarą, argumentuojantį įverčio pasirinkimą. Modeliui priskiriamas vaidmuo „Šv. Rašto ekspertas“, pagal tyrimus tai padidina vertinimo tikslumą [MPM+24]. Taip pat nurodoma turinį vertinti labiau nei gramatiką, nors ir klausimas sklandus, bet jeigu turinys neaiškus, tame didaktinės vertės nėra (žr. 12 pav.). Nurodytoje JSON struktūroje liepiama sukurti kiekvieno klausimo įverčio (angl. *grade*), komentaro (angl. *comment*) ir *id* laukus (žr. 9 pav.).

13 vartotojo užklausos dalyje įstatomas šaltinio skyriaus tekstas bei JSON klausimų sąrašas. Šaltinis yra būtinas, siekiant patikrinti klausimo ir Evangelijos skyriaus turinių atitikimą.

```
user_prompt = f"Biblijos ištrauka: {text}
Klausimai vertinimui: {questions_json_str}"
```

13 pav. *Klausimų vertinimo vartotojo užklausos dalis*

```
"metadata": {
    "evaluator_model": "gemini/gemini-2.5-flash",
    "source": "Jn_1"
},
"results": [
    {
        "id": "Jn_1_001",
        "grade": 5,
        "comment": "Klausimas yra aiškus, nuoroda į šaltinį
        tiksli, o atsakymas tiesiogiai atitinka pateiktą
        Biblijos ištrauką."
    }
]
```

14 pav. *Išsaugoto JSON klausimo įvertinimo pavyzdys*

Rezultatai

Atlikus 3688 sugeneruotų klausimų kryžminį įvertinimą, gautas 7376 individualių įvertinimų rinkinys. 3 ir 2 lentelėse pateikta suvestinė rodo koreliaciją tarp modelio pajėgumo ir generuojamo turinio tikslumo.

`Gemini-2.5-flash` pademonstravo beveik nepriekaištingą (99,7 %) gautų maksimalių įvertinimų statistiką, o`mistral-small` modelio rezultatai siekė tik (79,9 %). Modelis generavo paviršutiniškus arba nelogiškus klausimus, kuriuos kiti modeliai įvertino prastai. `Mistral-medium modelis` nuo laimėtojo atsiliko minimaliai, surinkdamas (97,1 %).

3 lentelė. *Sugeneruotų ir įvertintų klausimų rezultatai*

Modelis	**Sugeneruota**	**Įvertinta**	**Gauta įvertinimas 5**	**5 įvertintų klausimų sk.**
Gemini-2.5-flash	1226	2452	2444 (99,7 %)	1219
Mistral-medium-2508	1239	2478	2405 (97,1 %)	1169
Mistral-small-2506	1223	2446	1954 (79,9 %)	877
Iš viso	**3688**	**7376**	**6803** (**92,2 %**)	**3265**

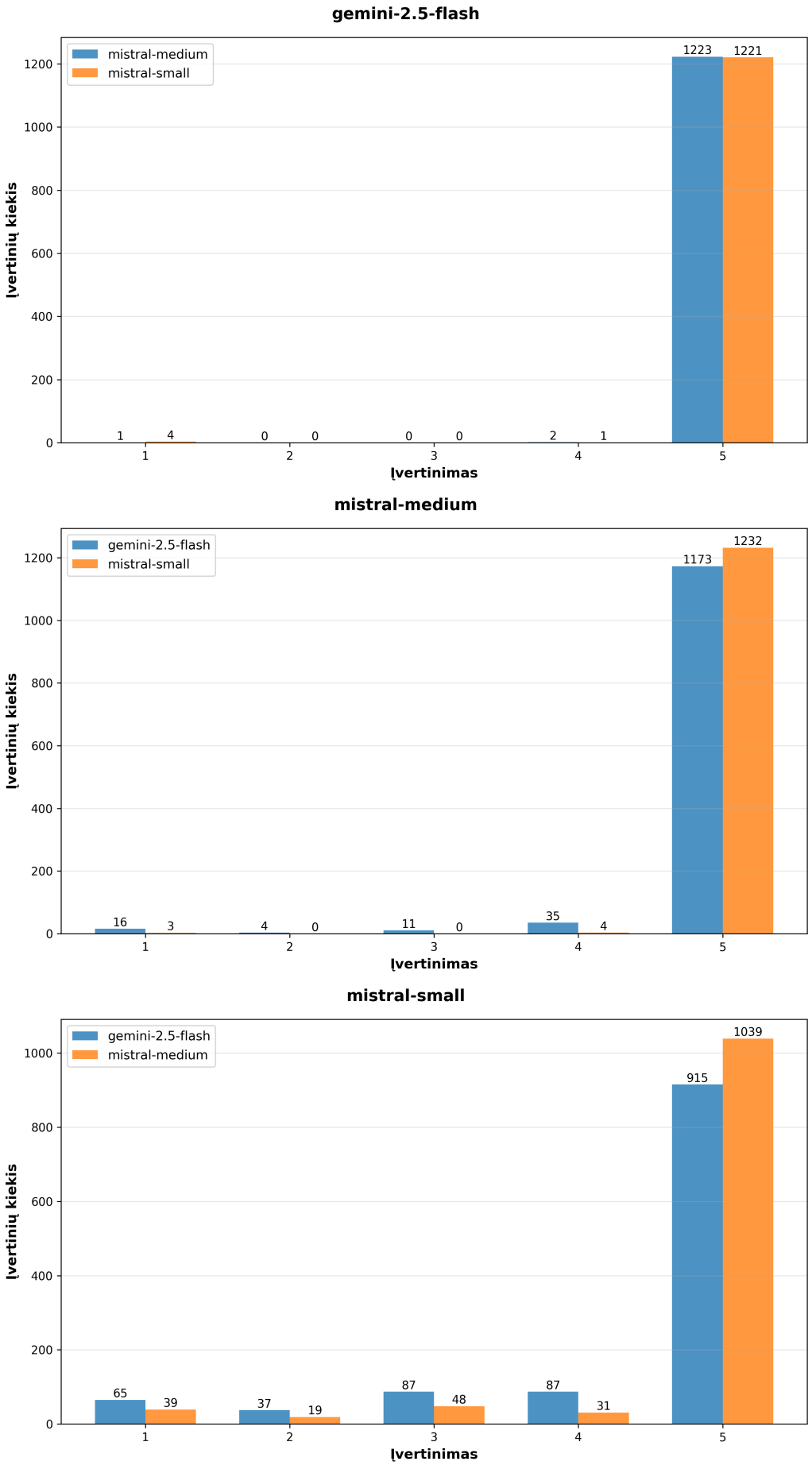
Žvelgiant į stulpelines modelių diagramas 15, matome, jog nė vienas iš 3688 sugeneruotų klausimų negavo įvertinimo 0. Reiškia visi modeliai sugeneravo prasmingus klausimus. Nors *Mistral-small* vienintelis turi matomą žemesnių balų (1–4) pasiskirstymą. Gemini-2.5-flash jį vertino griežčiausiai (tik 74,8 % gavo 5 balus). Tai veda prie išvados, jog stipresni modeliai geriau atpažįsta silpnesnių daromas klaidas, o silpnesni modeliai kaip tik vertina atlaidžiau. *Mistral-small* beveik visus kitų modelių klausimus vertino 5 balais, nesugebėdamas identifikuoti semantinių ir kitokių klaidų.

Analizuojant prastus generavimo atvejus, pasimatė dvi pagrindinės klaidų kategorijos:

1. **Semantinės klaidos:** *Mistral-small* modelis neskiria aplinkos detalių (pvz., „Dangaus“) nuo subjekto (pvz., „Dievo“) (žr. 16 pav.). Modelis sugalvojo teisingą atsakymą, kurio tekste nėra.

2. **Turinio haliucinacijos:** Modelis įterpia pagal prasmę derinčias frazes, bet neegzistuojančias šaltinyje. (žr. 17 pav.).

`Gemini-2.5-flash` užfiksavo smulkius faktinius šaltinio ir klausimo turinio neatitikimus, kurių silpnesni modeliai nepastebėdavo. Tai patvirtina, kad kryžminis vertinimas yra patikimas būdas klausimams atrūšiuoti. Tačiau, tyrimas taip pat patvirtino, jog LLM modeliai nėra deterministiniai - ta pati įvestis davė skirtingą rezultatą. Tai reiškia sugeneruotų klausimų skaičius ne visada atitiko užklausoje nurodytą. Pavyzdžiui, iš Jono evangelijos 6 skyriaus `Mistral-medium` sugeneravo 24 klausimus, o `Mistral-small` ir `Gemini-2.5-flash` po 23 klausimus. Rezultatai parodė, kad klausimų kiekio pokyčius reikia fiksuoti ir valdyti.



15 pav. *Kryžminio vertinimo balų pasiskirstymas pagal modelius*

```
Id: Mk_1_002
Klausimo autorius: mistral-small
Klausimo vertintojas: gemini-2.5-flash
Klausimas: Kas pasakė: „Tu mano mylimasis Sūnus, tavimi aš
tavimi aš gėriuosi"?
  a: Jonas Krikštytojas
  b: Dvasia
  c: Dangus (teisingas)
  d: Dievas
Įvertinimas: 2
Komentaras: Klausimas prašo nurodyti, kas pasakė frazę, o
pateiktas atsakymas „Dangus" (Heaven) yra problematiškas,
nes Dangus yra vieta, o ne kalbantis asmuo. Nors balsas
sklido iš Dangaus, kalbėtojas yra Dievas, todėl atsakymas
turi didelių turinio trūkumų ir klaidinančią logiką.
```

16 pav. *Pirmas prasto įvertinimo pavyzdys, kuriame vertinantis modelis pagrįstai aptinka klaidą ir prastai įvertina. Antras klausimas iš Morkaus Evangelijos pirmo skyriaus.*

```
Id: Mt_10_005
Klausimo autorius: mistral-medium
Klausimo vertintojas: gemini-2.5-flash
Klausimas: Ką reikėjo padaryti, jei namai, į kuriuos įeita,
**nebus verti** pasveikinimo? (Mt 10,13)
  a: Juos pasmerkti ir palikti prakeiksmo žodžius
  b: Nusikratyti dulkes nuo kojų, išėję iš namų
  c: Grįžti atgal ir bandyti dar kartą
  d: Jų nebeįžvelgti ir eiti toliau tyliai (teisingas)
Įvertinimas: 1
Komentaras: Nurodytame Biblijos tekste (Mt 10,13) nėra
nurodymo 'jų nebeįžvelgti ir eiti toliau tyliai', todėl
atsakymas faktiškai klaidingas.
```

17 pav. *Antras prasto įvertinimo ir haliucinacijos pavyzdys. Penktas klausimas iš Mato Evangelijos dešimto skyriaus.*

```
Id: Lk_18_010
Klausimo autorius: gemini-2.5-flash
Klausimo vertintojas: mistral-medium
Klausimas: Kokių įsakymų laikymąsi Jėzus priminė kilmingam turtuoliui?
  a: Mylėti Dievą ir artimą.
  b: Pasninkauti ir duoti dešimtinę.
  c: Nesvetimauk, nežudyk, nevok, melagingai neliudyk,
  gerbk savo tėvą ir motiną. (teisingas)
  d: Statyti šventyklas.
Įvertinimas: 5
Komentaras: Klausimas ir atsakymas idealiu visais
aspektais, atitinka turinį, logiką, gramatika ir didaktinę vertę.
```

18 pav. *Puikaus įvertinimo pavyzdys. Septintas klausimas iš Luko Evangelijos aštuoniolikto skyriaus.*

Išvados

Šiame darbe aptartos didžiausios Lietuvos švietimo sistemos problemos bei aprašyta kaip dirbtinio intelekto įrankiai gali padėti šiuos iššūkius spręsti. Kompetentingų pedagogų bei mokymosi medžiagos pritaikomumo spragas gali užpildyti individualiai prie mokinio prisitaikančios ir automatiškai turinį generuojančios LLM sistemos.

Eksperimentinio tyrimo metu autoriaus sukurta sistema geba automatiškai generuoti bei įvertinti klausimus pasitelkiant 3 LLM modelius (*Gemini-2.5-flash*, *Mistral-medium* ir *Mistral-small*). Rezultate sukurti, kryžmiškai įvertinti ir atrinkti 5 balus iš dviejų kitų modelių gavę **3265** klausimai. Šie klausimai yra tinkami tikrinti keturių Naujojo Testamento (Mato, Morkaus, Luko ir Jono) Evangelijų skaitymo atidumą katechezės veiklose. Kokybiškiausius bei labiausiai mokymui tinkamus klausimus generavo **Gemini-2.5-flash** modelis, net **99.7%** klausimų buvo įvertinti maksimaliai, penkiais balais. Tačiau svarbu neužmiršti, kad šis skaičius kilo iš modelių įvertinimų, todėl būtinas ateities žingsnis turėtų būti žmogaus eksperto auditas nedidelei daliai klausimų, leisiantis pilnai įvertinti LLM kryžminio klausimų vertinimo patikimumą. Tyrimas patvirtino, kad kryžminis vertinimas yra patikimas tik tada, kai vertintojo vaidmenį atlieka stipresnis modelis (pvz., *Gemini-2.5-flash*), kuris geba identifikuoti semantines, haliucionacijų ir kitokias modelių klaidas.

Tyrime sukurta automatinė klausimų generavimo sistema gali būti pritaikyta kuriant edukacinį turinį iš daugelio struktūrizuotų šaltinių (pvz., Senasis Testamentas). Žmogaus intervencija būtų reikalinga tik formuluojant LLM užklausas.

Šio tyrimo rezulatas iliustruoja, kad dirbtinio intelekto įrankiai jau dabar teikia pamatuojamą naudą ugdymo procesuose. Tačiau tyrimai ir sveika nuovoka primena, jog DI įrankiai kol kas neatstoja mokytojo, kuris geba palaikyti gyvą ryšį su mokiniu bei ugdyti jo kritinį mąstymą ir savarankiškumą.

3. Programinio kodo pateikimas

Siekiant užtikrinti rezultatų atkuriamumą, visas tyrimo programinis kodas yra viešai prieinamas *GitHub* saugykloje. Skaitytojas gali pasiekti visą projekto failų struktūrą. [Žal26].

Nuoroda į GitHub projekto saugyklą:

`https://github.com/vabalass/LLM-cross-examination-with-Bible/tree/main`

Literatūra ir šaltiniai

[ARA24] S. Al Faraby, A. Romadhony, Adiwijaya. „Analysis of LLMs for educational question classification and generation“. Iš: *Computers and Education: Artificial Intelligence* 7 (2024), puslapis 100298. https://doi.org/10.1016/j.caeai.2024.100298.

[CCD+25] A. Chatterji, T. Cunningham, D. Deming, Z. Hitzig, C. Ong, C. Shan, K. Wadman. „How People Use ChatGPT“. Iš: *OpenAI* (2025), puslapiai 1–63. URL: https://cdn.openai.com/pdf/a253471f-8260-40c6-a2cc-aa93fe9f142e/economic-research-chatgpt-usage-paper.pdf.

[Cen24] S. P. Center. *When Were the Gospels Written?* 2024. URL: https://stpaulcenter.com/posts/when-were-the-gospels-written.

[dra24] L. B. draugija. *Šventasis Raštas. Naujasis Testamentas (vertė kun. Č. Kavaliauskas, 1988 m. redakcija)*. 2024. URL: https://biblija.lt/index.php?id=3 (žiūrėta 2024-04-25).

[Gem25] Gemini Team, Google. *Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities*. Techninė ataskaita. Technical Report. Google DeepMind, 2025. URL: gemini-report@google.com.

[LBH15] Y. LeCun, Y. Bengio, G. Hinton. „Deep learning“. Iš: *Nature* 521.7553 (2015), puslapiai 436–444.

[Lit] LiteLLM. *LiteLLM*. URL: https://docs.litellm.ai/docs/ (žiūrėta 2026-01-03).

[LLH+23] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, P. Liang. „Lost in the Middle: How Language Models Use Long Contexts“. Iš: *Transactions of the Association for Computational Linguistics* 11 (2023), puslapiai 1097–1117.

[MMR+55] J. McCarthy, M. L. Minsky, N. Rochester, C. E. Shannon. „A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence“. Iš: *Dartmouth College / Research Proposal* (1955), puslapiai 1–13.

[MPM+24] S. S. Mucciaccia, T. M. Paixão, F. Mutz, A. F. De Souza, C. S. Badue, T. Oliveira-Santos. „Automatic Multiple-Choice Question Generation and Evaluation Systems Based on LLM: A Study Case With University Resolutions“. Iš: *LREC-COLING 2024* (2024), puslapiai 2246–2260.

[Pyt25] Python Software Foundation. *Python 3.13.1 Documentation*. 2025. URL: https://docs.python.org/3/ (žiūrėta 2026-01-18).

[Ric25] L. Richardson. *Beautiful Soup 4.13.0 documentation*. anglų. 2025. URL: https://www.crummy.com/software/BeautifulSoup/bs4/doc/.

[RJL+25] A. Rastogi, A. Q. Jiang, A. Lo, G. Berrada, G. Lample ir kiti. *Magistral*. Techninė ataskaita. Technical Report. Mistral AI, 2025. URL: https://huggingface.co/mistralai/Magistral-Small-2506.

[RN10] S. J. Russell, P. Norvig. *Artificial Intelligence: A Modern Approach*. 3rd. Upper Saddle River, New Jersey: Pearson Education (Prentice Hall), 2010. ISBN: 978-0-13-604259-4.

[Sho77] E. H. Shortliffe. „MYCIN: A Knowledge-Based Computer Program Applied to Infectious Diseases“. Iš: *Proceedings of the Annual Meeting of the Society for Computer Medicine*. Stanford University School of Medicine. Las Vegas, Nevada, 1977, puslapiai 66–69.

[SYD+26] Y. Shi, K. Yu, Y. Dong, F. Chen. „Large language models in education: a systematic review of empirical applications, benefits, and challenges“. Iš: *Computers and Education: Artificial Intelligence* 10 (2026), puslapis 16. URL: https://www.sciencedirect.com/science/article/pii/S2666920X25001699.

[Tur50] A. M. Turing. „Computing Machinery and Intelligence“. Iš: *Mind* 49 (1950), puslapiai 433–460.

[Wei66] J. Weizenbaum. „ELIZA - A Computer Program For the Study of Natural Language Communication Between Man And Machine“. Iš: *Communications of the ACM* 9.1 (1966), puslapiai 36–45.

[Žal26] B. Žalneravičius. *LLM cross-examination with Bible*. https://github.com/vabalass/LLM-cross-examination-with-Bible/tree/main. 2026.