

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Alex Vahter 213119IAIB

**SUURTEL KEELEMUDELITEL PÕHINEVA ÕPPEINFO
PÄRINGUTE JA ANALÜÜSI SÜSTEEMI ARENDAMINE**

Bakalaureusetöö

Juhendaja: Ago Luberg
PhD

Tallinn 2024

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Alex Vahter

25.05.2024

Annotatsioon

Suurtel keelemudelitel põhineva õppeinfo päringute ja analüüsi süsteemi arendamine

Käesoleva bakalaureusetöö eesmärgiks on arendada Tallinna Tehnikaülikooli õppejõududele ja abiõppejõududele süsteem, mis võimaldab koondada eri andmeallikatest infot kesksesse rakendusse ning selles andmestiku kohta päringuid ning analüütilisi küsimusi küsida.

Rakendus jaguneb esirakenduseks ja tagarakenduseks. Esirakendus on realiseeritud Discordi juturoboti kujul. Tagarakendus on kirjutatud Pythonis ning kasutab LlamaIndexi raamistikku rakenduse eri toimingute orkestreerimiseks, Ollamat keelemudelite halduseks ning vastuste genereerimiseks, Langfuse'i viipade ning metaandmete hoiustamiseks, Elasticsearchi andmete talletamiseks ning Kibanat talletatud andmete halduseks.

Täpsemalt võimaldab esirakendus kasutajatel saata süsteemi vajalikku teavet nii privaatsete kui ka avaliku kanali sõnumite kaudu. Tagarakendus töötleb automaatselt sisendandmed läbi ning talletab need vektorteisenditena andmebaasi. Loodud vektorteisendid võimaldavad rakendusel vastata keelemudelite abil keerulistele küsimustele ning seda tehes tugineda mitmele erinevale failile.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 28 leheküljel, 7 peatükki, 18 joonist, 0 tabelit.

Abstract

The Use of Large Language Models for Study Information Retrieval and Analysis

The aim of this bachelor's thesis is to develop a system for the faculty and teaching assistants of Tallinn University of Technology that consolidates information from various data sources into a central application where queries and analytical questions about the dataset can be posed.

The application is divided into a front-end and a back-end. The front-end is implemented as a Discord chatbot. The back-end is written in Python and uses the LlamaIndex framework to orchestrate various operations, Ollama for language model management and response generation, Langfuse for storing prompts and metadata, Elasticsearch for data storage, and Kibana for managing stored data.

Specifically, the front-end allows users to send necessary information to the system via both private and public channel messages. The back-end automatically processes the input data and stores it as embeddings in the database. These created embeddings enable the application to answer complex questions using language models, relying on multiple different files.

The thesis is in Estonian and contains 28 pages of text, 7 chapters, 18 figures, 0 tables.

Lühendite ja mõistete sõnastik

API	<i>Application programming interface</i> , rakendusliides, arvutisüsteemi teenuste andmise süntaksi ja semantika määratlus
CSV	<i>Comma separated values</i> , tekstifaili tüüp, mis sisaldab vorminguta andmeid- kirjeid eraldab reavahetuskoode, kirje välju eraldavad komad
Dokument	<i>Document</i> , Elasticsearchi JSONi objekt
Eeltöötlus	<i>Pre-processing</i> , toorandmestiku generatsioon masinõppe mudelite jaoks
EPUB	<i>Electronic publication</i> , e-raamatu failitüüp
GPT	<i>Generative Pre-trained Transformer</i> , masinõppe algoritmi liik, mis kasutab süvaõpet ning suurt andmestikku loodava mudeli treenimiseks ning uue viiba põhise vastuse genereerimiseks
Hetkvõte	<i>Snapshot</i> , ühe või mitme ettemääratud mäluala ühe ajahetke tõmmis
HTML	<i>Hypertext Markup Language</i> , hüpertexti märgistuskeel, veebilehtede loomise märgistuskeel
HWP	<i>Hangul Word Processor</i> , Hangul tekstitöötlustarkvara andmetüüp
Inverteeritud indeks	<i>Inverted index</i> , Elasticsearch dokumentidele viitav unikaalsete sõnade kogum
IPYNB	<i>Jupyter Notebook</i> andmetüüp
JPEG	<i>Joint Photographic Experts Group</i> , pildifaili vorming
JSON	<i>JavaScript Object Notation</i> , JavaScripti alamhulgal põhinev andmevahetusvorming
Kestvus	<i>Span</i> , Langfuse'i ühe vaatluse tööühiku kestvus
Kobar	<i>Cluster</i> , mingiks otstarbeks ühtseks tervikuks koondatud sarnaste objektide rühm
Konveier	<i>Pipeline</i> , andmetöötluselementide jada
Loome	<i>Generation</i> , Langfuse'i kestvuste liik uute masinõppe mudelite poolt genereeritud andmete talletamiseks
MBOX	Standardformaad sõnumite ning manuste talletamiseks

MD	<i>Markdown</i> , väikesemahuline lihtteksti vormindamise süntaks
MP3	Kadudega tihendamisega helifailivorming
MP4	Video- ja audioandmete kadudega tihendamise multimeediumikonteiner
Paindlikkus	<i>Flexibility</i> , süsteemi või komponendi kohandatavus kasutamiseks uutes rakendustes või keskkondades, mille jaoks see ei ole spetsiaalselt loodud
PDF	<i>Portable Document Format</i> , PostScriptil põhinev dokumentide vorming
PNG	<i>Portable Network Graphics</i> , rastergraafika vorming kadudeta tihendusega
PPTX	<i>PowerPoint Open XML Presentation</i> , Microsoft PowerPoint andmetüüp
RAG	<i>Retrieval Augmented Generation</i> , välistoeline genereerimine, keelemudelite, eriti semantilise otsingu, lähenemine, kus süsteem otsib ning rakendab olemasolevaid teadmisi uue sisu genereerimiseks ja päringutele vastete leidmiseks
RAM	<i>Random Access Memory</i> , muutmälu, suvapöördusmälu
REST	<i>Representational State Transfer</i> , esitusoleku siire, tarkvaraarhitektuuri laad, mis tagab teiste rakendustega koostoimimise
Sõlm	<i>Node</i> , tehniliste objektide kogum, mis moodustab paigutuseks üheainsa ruumilise üksuse
Tekstitükeldiste teisendused vektorruumis	<i>Embeddings</i> , tekstiühikute vektorteisendid
Sõnestus	<i>Tokenization</i> , analüüsitava tekstivoo tükeldamine sõnedeks
Vaatlus	Langfuse'i ühe päringu või operatsiooni ühik, mis sisaldab tavaliselt funktsiooni algsidendit ja lõppväljundit
Viip	<i>Prompt</i> , suurte keelemudelite vastuse suunamiseks kohandatud käsk
XLSX	<i>Microsoft Excel Spreadsheet</i> , Microsoft Excel tabelitöötlustarkvara andmetüüp

Sisukord

1	Sissejuhatus	9
1.1	Töö skoop	9
2	Olemasolevad süsteemid ning rakendused	11
2.1	Ülikoolis kasutusel olevad lahendused ja arhitektuuri ülevaade	11
2.1.1	Olemasolevate teenuste majutus	11
2.1.2	Microsoft Excel	12
2.1.3	Google Sheets	12
2.1.4	Discord	13
2.2	Võimalikud alternatiivsed terviklahendused	13
2.2.1	Copilot for Microsoft 365	13
2.3	Ülevaade võimalikest rakenduses kasutatavatest komponentidest	13
2.3.1	Elastic Stack	14
2.3.2	OpenSearch	14
2.3.3	Haystack	15
2.3.4	LlamaIndex	15
2.3.5	LangChain	16
2.3.6	Langfuse	16
2.4	RAG vs mudeli peenhäälestus	17
2.4.1	Välistoeline genereerimine	17
2.4.2	Mudeli peenhäälestus	18
3	Arendustöö nõuded	19
3.1	Funktsionaalsed nõuded	19
3.2	Mittefunktsionaalsed nõuded	19
4	Kasutatavad tehnoloogiad	20
4.1	Projekti arhitektuur	20
4.2	Langfuse	21
4.3	LlamaIndex	21
4.4	Ollama	22
4.5	Discord	22
4.6	Elastic Stack	22
5	Arendusprotsess	23
5.1	Rakendatavad arendusvõtted	23

5.2	Elastic Stack seadistused	23
5.2.1	Elasticsearch	23
5.2.2	Kibana	24
5.3	Ollama seadistused	24
5.4	Rakenduse paigaldus ning käivitus	25
5.5	Langfuse	26
5.5.1	Päringute logimine	26
5.5.2	Viibahaldus	27
5.5.3	Statistika	28
5.6	Sobiva suure keelemudeli valik	29
5.6.1	Gemma	29
5.6.2	Llama2	30
5.7	Discord	31
6	Tulemused	34
6.1	Loodud funktsionaalsused	34
6.2	Edasiarenduse võimalused	34
7	Kokkuvõte	36
	Kasutatud kirjandus	37
	Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	41
	Lisa 2 – Langfuse Trace'i sisu	42

Jooniste loetelu

1	Süsteemi arhitektuuri ülevaade.	21
2	Kibana dokumendivaade.	24
3	Langfuse'i <i>Trace Detail</i> vaade.	27
4	Langfuse'i <i>Prompts</i> vaade.	28
5	Langfuse'i <i>Dashboards</i> vaade.	29
6	Juturoboti koodilõigu genereerimine.	30
7	Juturoboti vastusekvaliteedi näide.	31
8	Juturoboti sisendandmete laadimine.	32
9	Juturoboti vastuse tagasisidestamine.	33
10	Juturoboti vastuse genereerimise staatus.	33
11	Langfuse <i>Trace Detail LlamaIndex query</i> vaade.	42
12	Langfuse <i>Trace Detail query</i> vaade.	43
13	Langfuse <i>Trace Detail retrieve</i> vaade.	43
14	Langfuse <i>Trace Detail HuggingFaceEmbedding</i> vaade.	44
15	Langfuse <i>Trace Detail reranking</i> vaade.	44
16	Langfuse <i>Trace Detail synthesize</i> vaade.	45
17	Langfuse <i>Trace Detail templating</i> vaade.	45
18	Langfuse <i>Trace Detail Ollama llm</i> vaade.	46

1. Sissejuhatus

Tallinna Tehnikaülikoolis on kasutusel erinevates õppeainetes mitmesugused keskkonnad ning abivahendid, mille abil hoiustada, otsida ning jälgida õppetöö läbiviimiseks vajalikku infot. Laialdaselt on kasutusel näiteks Moodle'i platvorm õppetöö ning tudengite hinnete halduseks. See võimaldab iga kursuse õppejõududel saada ülevaadet nii tudengi isiklikest tulemustest kui ka tudengite keskmistest tulemustest ülesannete kaupa. Antud lahenduse üheks probleemiks on aga suur andmehulk ning vähesed tööriistad andmete pärimiseks ning analüüsi läbiviimiseks, mille tagajärjena on keerulisem tuvastada mahajäänud õppureid ning tuvastada murekohti kursuse läbiviimisel. Selliste probleemide varajasem märkamine ning nende põhjuste kiirem tuvastamine võimaldaks kohandada kursust vastavalt vajadusele või pakkuda tudengitele personaalsemat tagasisidet ja tuge, vähendades läbikukkujate osakaalu.

Probleemi lahendamiseks on hetkel õppejõududel võimalus viia analüüs käsitsi läbi, kasutades neile teada ja tuntud vahendeid näiteks tabelitöötlustarkvarasid nagu Microsoft Excel ning Google Sheets. Nende vahenditega on võimalik luua põhjalikum statistika ning visualiseerida õppijate sooritusi kursuse raames. Kahjuks on selline manuaalne lähenemine ajakulukas ning vajab andmestiku sügavat mõistmist. Lisaks ei ole võimalik mugavalt seostada üht andmestikku teistega nagu õppetulemused ning ülesannete sisu. Selline analüüs vajaks põhjalikumat teadmist kasutatavatest tööriistadest ning on teostatav ainult kogenumatel oma ala professionaalidel.

Lõputöö eesmärgiks on arendada pilootrakendus, mis rakendaks suuri keelemudeleid õppeinfo sügavamaks ning mugavamaks analüüsiks ning kiirendaks murekohtade tuvastusprotsessi. Sarnaselt olemasoleva töövooga on võimalik õppejõududel neile aktuaalsetest süsteemidest koguda kokku uuritav andmestik ning see edastada loodud juturobotile, mis töötleb sisendandmed sobivale kujule ning vastab erinevatele päringutele sisendandmetest lähtuvalt, minimeerides analüüsile kuluvat aega. Lisaks võimaldab juturobot seostada mitme erineva sisendallika andmed, sõltumata nende originaalkujust ning vastata koondatud andmetele põhinedes.

1.1 Töö skoop

Projekti pilootrakenduse printsiibist lähtudes on esialgseks kasutusjuhiks ning sihtgrupiks Tallinna Tehnikaülikooli infotehnoloogia teaduskonna "Programmeerimise algkursuse"

õppejõud ning abiõppejõud.

Rakenduse arendusel on üheks põhifookuseks seatud eri platvormide ja keskkondade agnostilisus ning skaleeritavus, et oleks võimalik soovi korral rakenduse paigalduskeskkonda, kasutatavaid keelemudeleid ning sisendandmestikku kohandada ning muuta vastavalt vajadusele. Sellest fookusest lähtuvalt ei ole süsteem limiteeritud vaid ühele õppeainele ega kasutusjuhule.

2. Olemasolevad süsteemid ning rakendused

Uue süsteemi kavandamisel on vajalik kaardistada olemasolevad ning aktiivses kasutuses süsteemid. Lisaks on vajalik välja selgitada võimalikult optimaalne töövoog ning kaardistada funktsionaalsused, mis loovad võimalikult palju lisaväärtust, kuid ei tekita kasutajatele lisakeerukust.

2.1 Ülikoolis kasutusel olevad lahendused ja arhitektuuri ülevaade

Järgnevalt analüüsitakse Tallinna Tehnikaülikooli infotehnoloogia teaduskonna "Programmeerimise algkursuse" korralduses kasutusel olevaid lahendusi tudengite õppeedukuse ja -tulemuste analüüsimiseks ning õppeinfo halduseks ja nende võimalikke alternatiive ning ka arhitektuurseid valikuid.

2.1.1 Olemasolevate teenuste majutus

Tallinna Tehnikaülikoolis on autorile teadaolevalt kasutusel nii pilvteenusepakkujad nagu Microsoft Azure kui ka organisatsioonisisised serverid erinevate süsteemide jaoks.

Pilvteenuste üheks märkimisväärseks eeliseks on skaleeritavus ning paindlikkus. See aitab tagada teenuste optimaalse toimimise vaatamata kasutajate hulgale ning süsteemi koormusele. Pilvteenuste skaleeritav ülesehitus ning jagatav arhitektuur võimaldab pakkuda teenuseid tellimuse- ja nõudluspõhiselt, mistõttu on võimalik maksta vaid nende ressursside eest, mis tegelikult kasutust leiavad. See võimaldab kulusid minimeerida ning vähendada broneeritavate ressursside hulka vähese kasutusega perioodidel ning suurendada suure nõudlusega perioodidel. Pilvteenused tagavad ka parema kättesaadavuse erinevate tehniliste probleemide esinedes, sest teenusepakkuja infrastruktuur on loodud võimalikult veakindlana. Sellega kaasneb ka lihtsam taastevõime ning suurendab süsteemide vastupidavust ootamatutes olukordades. [1]

Pilvteenuste miinusteks on sõltuvus internetiühendusest, mistõttu ei ole võimalik teenuseid lokaalselt organisatsioonides rakendada ning teenuste kiirus sõltub suuresti organisatsiooni välisvõrgu internetikiirusest. Meeles tuleb pidada ka kohalikke regulatsioone andmete hoiustamisega, sest andmed asuvad kolmanda osapoole hallatavates serverites. [2]

Erinevalt pilvteenuste pakkujate serveritest, võimaldavad lokaalsed serverid

organisatsioonidele täielikku kontrolli hoiustavate andmete ja infrastruktuuri üle, mis tagab suurema kontrolli turvalisuse ja privaatsuse üle, sest süsteemid asuvad füüsiliselt ettevõtte ruumides. Samuti võimaldavad lokaalsed serverid teatud olukordades suuremat jõudlust, sest puudub sõltuvus välisest internetikiirusest. [3]

Lokaalsete serverite puuduseks on suur esialgne süsteemi soetamis- ning paigalduskulu ning pidevad ülalpidamiskulud hooldustööde ja -personali vallas. Lisaks on vajalik lisahoolsus andmete varundamisel ning taastamisel tehniliste probleemide esinedes. [4]

Rakenduse paigaldamisel ning keskkonna valikul lähtutakse eelnevalt välja toodud punktidest, töö skoobist ning sätestatud prioriteetidest.

2.1.2 Microsoft Excel

Microsoft Excel on tasulisse Microsoft 365 komplekti kuuluv tabeliarvutustarkvara, millele on nii õppejõududel kui ka üliõpilastel tasuta ligipääs Tallinna Tehnikaülikooli Microsofti kontode abil. Excel võimaldab olemasolevatest andmestikest koostada erinevaid analüüse, diagramme ning graafikuid, et tagada andmestiku parem mõistmine. Excelisse on võimalik importida nii CSV (*Comma-separated values*), XLSX (*Microsoft Excel Spreadsheet*), HTML (*HyperText Markup Language*) kui ka tekstifaile [5]. Võimalik on loodud Exceli töövihikuid jagada ka teiste Microsofti kontoga kasutajatega ning samaaegselt koos töötada. Excelis on võimalik ka erinevate makrode ning skriptide loome, mis võimaldab keerukamate protsesside läbiviimist kui ka töövoogude optimeerimist. [6]

Exceli üheks võimalikuks miinuseks on kasutajamugavus, nimelt on saadaval palju võimalikke tööriistu, mis lisab küll lisavõimekust, kuid muudab õigete tööriistade leidmise keerulisemaks. Üheks võimalikuks ohukohaks koos töötamisel on faili koopia jagamine, mitte faili enda jagamine, mistõttu võib tekkida kaks eraldi faili erineva infoga. Lisaks eeldab Exceliga koos töötamine, et kõigil osapooltel on kättesaadav vastav litsents Exceli kasutuseks. [7]

2.1.3 Google Sheets

Google Sheets on sarnaselt Microsoft Exceliga tabeliarvutustarkvara, mis kuulub Google'i kontoritarkvara komplekti. Erinevalt Excelist, on Sheetsis loodud failid kättesaadavad kõigile Google'i konto omanikele tasuta. Võimalik on koos töötada ka inimestega, kellel puudub konto, kuid neil puudub võimalus uut dokumenti luua ning selleks on siiski vajalik Google'i konto olemasolu. See võimaldab koostööd ka ülikooliväliste partneritega. [8]

Kuigi Sheets on kasutajasõbralikum ning ei eelda litsentse ega ka töölaua rakenduse olemasolu, puuduvad selles osad Exceli võimekused, mistõttu on kohati õpiandmete analüüsi koostamine keerulisem. Sarnaselt Exceliga leidub selles osaline tugi andmete automaatseks uuendamiseks ning analüüsiks, kuid kahjuks on antud funktsionaalsus suhteliselt pinnapealne ning ei paku vajalikku taset täielikuks õpiandmete halduseks. [9]

2.1.4 Discord

Discord on sotsiaalmeediaplattform, kus on võimalik tasuta luua eraldi alaservereid ning nendes kanaleid suhtluse organiseerimiseks [10]. Discord on kasutusel nii õppejõudude kui ka abiõppejõudude omavaheliseks suhtluseks ja tudengitega infovahetuseks. Serverites on kasutajatele määratud eraldi rollid, mis määravad nende ligipääsuastme ning tagab kasutajate lihtsa tuvastuse. Kasutusel on ka eraldi minirakendus kasutajatele rolli määramiseks.

2.2 Võimalikud alternatiivsed terviklahendused

Järgnevalt analüüsitakse ning antakse ülevaade planeeritava süsteemi ning arendustöö võimalikest olemasolevatest alternatiiv- ja terviklahendustest.

2.2.1 Copilot for Microsoft 365

Copilot for Microsoft 365 on Microsofti poolt pakutav lahendus, mis ühildub olemasolevate Microsoft 365 komplektidesse kuuluvate rakendustega nagu Teams. Copilot kasutab oma toimingute läbiviimiseks Microsoft Azure'is leiduvaid uusimaid Open AI keelemudeleid nagu GPT-4, mis tagab kiire ning võimsa andmepäringu ning -otsingu.

Vaatamata Copiloti sügavale integratsioonile Microsoft 365 komplektiga, vajab see eraldi litsentsi igale kasutajale, kes seda kasutada soovib. Vajalik litsents on aastapõhise lepinguga, mis piirab kasutajate valikuvõimalust ning lukustab kasutajad pikemaajaliselt valitud süsteemi. [11] Lisaks ei ole toimivus ning ühilduvus Microsofti-väliste teenustega garanteeritud.

2.3 Ülevaade võimalikest rakenduses kasutatavatest komponentidest

Järgnevalt analüüsitakse ning antakse ülevaade planeeritava süsteemi ning arendustöö võimalikest komponentidest.

2.3.1 Elastic Stack

Elastic Stack tarkvarakomplekt, tuntud ka kui ELK Stack, on lihtsalt skaleeritav ning sobilik suurtest andmehulkadest info otsingu ning analüüsiga tegelemiseks. See koosneb neljast põhikomponendist: Elasticsearch, Kibana, Logstash ja Beats. [12] Antud projekti raames oleksid vajalikud Elasticsearch ning Kibana.

Elasticsearch on eelkõige otsingumootor, mis on optimeeritud suurte ning keeruliste andmestikega töötamiseks. See rakendab inverteeritud indeksite (ingl. k. *inverted index*) lähenemist, mis muudab päringutele vastete leidmise kiireks. Elasticsearch on ka suurel määral skaleeritav ning paindlik. [13]

Kibana on Elasticsearchiga lähedalt põimitud analüütikatööriist, mis võimaldab Elasticsearchis leiduvate andmete kohta päringuid teha, neid hallata, visualiseerida ning analüüsida. Kibanas on võimalik koostada erinevaid töölaudu interaktiivsete graafikutega, mis annavad andmetest põhjaliku ülevaate. [12]

Elastic Stack on saadaval nii tasulisel kui ka tasuta kujul sõltuvalt vajadusest ning lisafunktsionaalsuste soovidest. Kogu tarkvarakomplekt on suurel määral kohandatav ning süvitsi omavahel integreeritud [14]. Lisaks on sellel suhteliselt suur kasutajaskond, mis muudab erinevate liidestuste leidmise eri süsteemidega lihtsamaks, tagades süsteemi paindlikkuse ning laiendusvõime.

2.3.2 OpenSearch

OpenSearch on 2021. aastal loodud vabavaraline haru Elasticsearchist ja Kibanast, mille fookuseks on turvalise, avatud ning kõrge kvaliteediga andmete töötluse ja hoiustuse platvormi arendus. Sarnaselt Elastic Stack tarkvarakomplektiga, leidub OpenSearch komplektis mitmeid alakomponente: OpenSearch, OpenSearch Dashboards ja Data Prepper [15]. Antud projekti raames ei ole Data Prepper praegusel ajahetkel vajalik.

OpenSearch ja OpenSearch Dashboards peamiseks eesmärgiks on andmete hoiustamine ning visualiseerimine. Need põhinevad viimsel avatud lähtekoodiga Elasticsearchi ja Kibana versioonidel, mistõttu leidub neis mitmeid ühiseid funktsionaalsusi. OpenSearchi levikut on kiirendanud suurte keelemudelite kasutuse kasv ning nende andmestike hoiustamisnõudlus. [16]

OpenSearchi suureks eeliseks on ka süsteemi arhitektuurne pool, kus süsteem on jaotatud

sõlmedeks (ingl. k. *node*) ning kobarateks (ingl. k. *cluster*), mis aitab tagada süsteemi töökindluse ka riistvaralise rikke puhul. [17]

2.3.3 Haystack

Haystack on Deepseti poolt arendatud avatud lähtekoodiga raamistik, mille eesmärgiks on lihtsustada keelemudelite rakendamist eri andmestike ja infoga. See võimaldab efektiivset ning kohandatavat infootsingut nii struktureeritud kui ka struktureerimata andmetega. [18]

Haystackil on liideseid nii Elasticsearchi, OpenSearchi kui ka erinevate SQL andmebaaside jaoks, mis muudab raamistiku integreerimise võimalikult lihtsaks ning paindlikuks. Haystack toetab ka andmestiku töötlemist dokumentideks ning toiminguid failidega nagu nende eeltöötlus, sõnestus, indekseerimine kui ka nende analüüs. [19]

Haystacki üheks eeliseks on avatud lähtekood, mis võimaldab raamistikku vajadusepõhiselt kohandada. Antud kohandatavuse suureks potentsiaalseks miinuseks Haystacki arhitektuuriga võib osutada aga koodi suurem keerukus ning raskem hallatavus.

Kuigi Haystack omab suurt kasutajaskonda, on selle integratsioonide arv eri konkurentidega võrreldes pigem tagasihoidlik. Lisakeerukust lisab ka Haystacki migratsioon Haystack 1.x versioonilt Haystack 2.0 versioonile, mistõttu on vajalik mitmete integratsioonide ning Haystack 1.x versioonidel põhinevate rakenduste ümbertegemine ning refaktoreerimine [20]. Suurte raamistiku muudatustega vananevad ka kasutajaskonna poolt loodud juhendid, mis raskendab arendusprotsessi märgatavalt.

2.3.4 LlamaIndex

Sarnaselt Haystackiga on LlamaIndex avatud lähtekoodiga andmetöötlusplatvorm ja raamistik, mille eesmärgiks on olla vahelüli rakenduse ja suurte keelemudelite vahel. LlamaIndex omab ka liideseid nii Elasticsearchi, OpenSearchi kui ka erinevate andmebaaside jaoks ning toetab toiminguid eri failidega nagu nende eeltöötlus, sõnestus, indekseerimine kui ka nende analüüs. [21]

LlamaIndexi suureks eeliseks on selle kogukond ja avatud lähtekood, mis julgustab erinevaid arendajaid ja teadlasi panustama platvormi arengusse ja selle parendamisse. LlamaIndexi ning LlamaHubi modulaarne ülesehitus võimaldab soovijatel luua näiteks uusi sisendandmete-, andmetöötlus- ning väljundandmetemoduleid. Lisaks võimaldab modulaarne lähenemine hoiduda lähtekoodi kohandamisest, mis aitab vältida koodi

keerukuse kasvu, suunates arendajaid moodulite arenduse suunas. [22]

2.3.5 LangChain

LangChain on suletud lähtekoodiga andmetöötlusplatvorm, mis keskendub suurte keelemudelite integreerimisele eri rakendustesse. LangChain pakub tasuta limiteeritud kasutust üksikarendajale, kuid sõltuvalt kasutusjuhust on tegu siiski enamasti tasulise teenusega. Sarnaselt LlamaIndexi ja Haystackiga omab LangChain kogukonda arendajatest, kes aitavad luua liideseid erinevate platvormidega ühendamiseks. Oma tasulise ülesehituse tõttu on LangChaini kasutajaskond küll väiksem, kuid see-eest on asetatud rohkem rõhku kasutajamugavusele, mis loob LangChaini integreerimise oma rakendusse mugavamaks. LangChain on lähedalt seotud ka LangSmithi ja LangServe'iga. [23]

LangSmithi abil on võimalik kiiresti ja mugavalt arendada keelemudelitel põhinevaid rakendusi ning pakub mugavat kasutajaliidest päringute logimiseks, metaandmete ja teenuse kiiruse jälgimiseks. LangSmith võimaldab ka rakenduse kiiret testimist testandmestikuga, et arendusprotsessi vältel jälgida, kuidas muudatused vastuse kvaliteeti ning kiirust mõjutada võivad. LangSmith on tasuline teenus, mis on hinnastatud vastavalt vaatluste (ingl. k. *trace*) arvule. [24]

LangServe võimaldab koostöös LangChaini ja LangSmithiga kiirelt ning mugavalt rakendusi API (ingl. k. *Application programming interface*) serveritena nende pilvteenusesse paigaldada. See toimib konveierite (ingl. k. *pipeline*) põhimõttel ehk ühendub koodivaramuga ning peale esialgset seadistust paigaldab muudatused automaatselt [25]. Sarnaselt LangChaini ning LangSmithiga on tegu tasulise lisateenusega.

2.3.6 Langfuse

Langfuse on 2023. aasta alguses asutatud vabavaraline suurte keelemudelite jälgimisteenus, mis on saadaval nii tasulisel kui ka tasuta kujul. Teenust on võimalik täiel mahul paigaldada kohalikule masinale kahe Docker konteineri kujul või kasutada Langfuse'i poolt pakutavat pilvteenust, mis on hinnastatud sõltuvalt vaatluste arvust kuus. Pilvteenuse tasuta pakett piirdub 50 000 vaatlusega kuus ning kõik, mis üle selle, on vastavalt Langfuse'i hinnakirjale. [26]

Langfuse on ettevõtteks suhteliselt noor, kuid see-eest on oma läbipaistvusega silma paistnud ning enda ümber koondanud suure hulga arendajaid, mistõttu areneb teenus kiiresti ning kasutajaskonnale vajalikul moel.

Langfuse sarnaneb funktsionaalsuse poolest LangSmithiga, kus on võimalik kõiki päringuid salvestada ning neid analüüsida, pakkudes eraldi töölauda keelemudelite hinnastuse, teenuse toimimise ning ka trendide jälgimiseks. Lisaks on võimalik hallata viipasid (ingl. k. *prompt*) otse Langfuse'i kasutajaliidesest ning märgistada vaatlused vastavate andmetega, et arendusprotsessi käigus muutustel silma peal hoida [27]. Langfuse'i kasutajaliideses on võimalik viia läbi ka katseid erinevate testandmestikega ja keelemudelitega ning on ka tugi erinevate testimisraamistike jaoks nagu Ragas. Testimisraamistikelt saadud tulemused seostatakse automaatselt vaatlustega, mis annavad objektiivse hinnangu vastuste kvaliteedist. [28]

Langfuse on lihtsasti paigaldatav pea kõikidesse keelemudeleid rakendavatesse süsteemidesse, omades logimiseks ning süsteemiga infovahetuseks nii Pythoni kui ka JavaScripti tuge. Lisaks toetab Langfuse ka süvitsi LangChaini ja LlamaIndexit.

2.4 RAG vs mudeli peenhäälestus

Masinõppes ning generatiivsete mudelite rakendamisel eri süsteemides on mitmeid võimalikke lähenemisi ning rakendusviise. Nendest kaks kõige populaarsemat on välistoeline genereerimine ning mudelite peenhäälestamine.

2.4.1 Välistoeline genereerimine

Välistoeline genereerimine ehk RAG (ingl. k. *Retrieval-Augmented Generation*) on keelemudelite, eriti semantilise otsingu, lähenemine, kus süsteem otsib ning rakendab olemasolevaid teadmisi uue sisu genereerimiseks ja päringutele vastete leidmiseks. See ühendab endas kaks peamist komponenti: informatsiooni hanke (ingl. k. *Retrieval*) ning sisu genereerimise (ingl. k. *Generation*). [29]

Informatsiooni hanke samm keskendub asjakohase teabe leidmisele andmestikust. Selle efektiivsus mõjutab suurel määral järgmise, sisu genereerimise, sammu kvaliteeti. Asjakohase teabe leidmisel on kasutusel sageli vektorandmebaasid, kuhu on lisatud varasemalt andmestikus leiduva info tekstitükeldiste teisendused vektorruumis (ingl. k. *embeddings*). Need on loodud andmestiku info tükeldamisega vastavalt sätestatud parameetritele ning loodud teisendus asetatakse vektorruumi semantiliselt sarnaste tekstiühikute vahetusse lähedusse. See teisendus aitab süsteemil sooritada otsinguid, keskendudes rohkem konteksti tähendusele, mitte seal leiduvatele märksõnadele. Loodud vektorandmebaasi poole pöörduvatele päringutele vaste leidmiseks, arvutatakse päringu tekstiühikutest vektorvasted ning kasutatakse algoritme nagu koosinus sarnasust (ingl. k.

cosine similarity) või skalaarkorrutist (ingl. k. *dot product*), lähimate semantiliste vastete leidmiseks. [30]

Sisu genereerimise samm rakendab varasemalt päritud teavet uue, arusaadava sisu loomeks. Selleks edastatakse mudelile nii viip, mis annab mudelile juhiseid, millist sisu ning tulemit oodatakse, kui ka vektorandmebaasist päritud teave, mille põhjal sisu luua. Info loomeks kasutatavaks mudeliks on sageli GPT (*Generative Pre-trained Transformer*), mis on loodud efektiivseks paralleelseks töötluseks ning konteksti tajumiseks suurtes andmehulkades [31]. Need GPT mudelid on transformer arhitektuuril põhinevad mudelid, mis on treenitud erinevatele andmemassiividele tuginedes. Treeningu vältel õpivad need mudelid keelelise konteksti mõistmist, et arvestada sõnadevaheliste seostega ning varasemale infole tuginedes ennustada järgmist sõna. See tekstiühikutepõhine vastusegenereerimine võimaldab vastust kasutajale striimida ehk edastada tekstiühiku haaval.

2.4.2 Mudeli peenhäälestus

Mudeli peenhäälestus (ingl. k. *fine-tuning*) on protsess, kus eelnevalt treenitud masinõppe mudelit kohandatakse vajadusepõhiselt uue väiksema ülesandepõhise andmestikuga. Peenhäälestus võimaldab mudelil säilitada suure andmestiku omadused ning teadmised, kuid neid paremini rakendada soovitud ülesande lahendamiseks. [32]

Mudeli peenhäälestamise võimalikeks riskideks on potentsiaalne mudeli ületreenimine, mistõttu mudel kohaneb liigselt uue ülesandepõhise andmestikuga ning võib kaotada oma varasemad teadmised või liigselt kinnistuda uude andmestikku, langetades andmestikuväliste küsimuste vastuste kvaliteeti. Lisaks on mudeli peenhäälestamine aja- ja ressursikulukas, mistõttu on mudeli uuendamine ja vahetamine raskendatud.

Peenhäälestatud mudelid on enamasti kasutuses staatiliste ning domeenispetsiifiliste andmestike põhjal päringutele vastamisel, nende vastuste kvaliteet ning erilise sõnavara ning konteksti valdamine on tavakasutusjuhtudel parem kui ainult RAGi kasutataval mudelitel. [33]

3. Arendustöö nõuded

Käesolevas peatükis on loetletud loodava rakenduse nõuded, tuginedes eelnevas peatükis teostatud analüüsile. Nõuded ning projekti skoop on määratud arendustöö lõppkasutaja ootustest ning olemasolevate lahenduste puudujääkidest. Nõuded on jaotatud kaheks: funktsionaalsed ning mittefunktsionaalsed nõuded.

3.1 Funktsionaalsed nõuded

Funktsionaalsed nõuded kirjeldavad süsteemis läbiviidavaid toiminguid ning võimalikke töövooge. Mainitud nõuete alla kuulub näiteks süsteemis leiduva info haldus. Antud süsteemile on sätestatud järgnevad funktsionaalsed nõuded:

- Rakenduse abil on võimalik pärida õppeaine läbiviimiseks vajalikke materjale.
- Rakenduses väljastatav info on usaldusväärne ning andmetest lähtuv.
- Rakendusse lisandub info vajadusepõhiselt ning andmeid on võimalik nii lisada kui ka eemaldada.

3.2 Mittefunktsionaalsed nõuded

Mittefunktsionaalsete nõuete alla kuuluvad süsteemi ülesehitust kirjeldavad nõuded, mis aitavad tagada funktsionaalsete nõuete täitmise. Nende alla kuuluvad näiteks süsteemi turvalisus, töökindlus, hallatavus ning skaleeritavus. Antud süsteemi mittefunktsionaalsete nõuete alla kuuluvad:

- Süsteemi lihtne hallatavus ning lisafunktsionaalsuse lisamist lubav arendusstiil.
- Süsteemi koodibaas on struktureeritud ning üles ehitatud valdkonna nõuetele vastavalt.
 - Koodibaasi keeleks on inglise keel.
 - Rakendatud on standardseid meetodeid ning stiili.
- Kasutajakogemuse kujundamisel tuleb lähtuda ligipäätavuse printsiipidest, et rakendus oleks kasutatav erinevate vajaduste ja võimalustega kasutajate poolt.
 - On arvestatud ekraanilugerite ning muude ligipääsutööriistade toimimisega.
- Süsteem peab olema kohandatav ning ülejäänud süsteemidest isoleeritud, et tagada töökindlus.
- Andmed on turvaliselt hoiustatud.

4. Kasutatavad tehnoloogiad

Käesolevas peatükis on võimalik saada ülevaade projektis kasutatavatest tehnoloogiatest, mida on kasutatud süsteemi arendamisel.

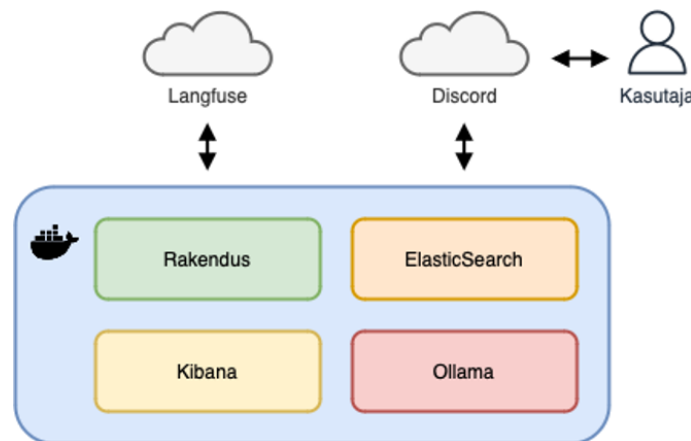
4.1 Projekti arhitektuur

Käesoleva bakalaureusetöö põhifookuseks on luua süsteem, mis oleks võimeline vastama sisendandmete põhjal keerukamatele sisulisele küsimustele.

Selle põhiosadeks on “Luciferi” nimega Discordi juturobot, mis rakendab Llamaindexit päringutele vastuse otsimiseks, Ollamat suurte keelemudelitega vastuse genereerimiseks, Langfuse'i viipade hoiustamiseks ning päringutega seostuvate metaandmete ja logide talletamiseks. Loodud süsteemi andmebaasina on kasutatud tarkvarakomplekti Elastic Stack. Täpsemalt on kasutuses Elasticsearchi andmete hoiustamiseks ning pärimiseks ja Kibana keskkonda andmestiku visualiseerimiseks ning selle haldamiseks. Andmevahetus rakenduse erinevate osade vahel on kujutatud joonisel 1.

Antud rakenduse sisendandmestik on plaanitud ning testitud txt lihttekstivormingu ning *Markdown* andmetüüpidega, tugi on ka loodud järgnevatele andmetüüpidele: CSV (*comma-separated values*), DOCX (Microsoft Word), EPUB (e-raamatu formaat), HWP (Hangul Word Processor), IPYNB (Jupyter Notebook), JPEG (JPEG pildiformaat), MBOX (*MBOX email archive*), MD (*Markdown*), MP3, MP4 (audio ja video), PDF (*Portable Document Format*), PNG (*Portable Network Graphics*), PPT, PPTX (Microsoft PowerPoint). Eelnevalt mainitud andmetüübid tulenevad LlamaIndexi SimpleDirectoryReader teegist, mis on kasutusel sisendandmete vektorandmebaasi teisendite loomisel [34]. Tulevikus on võimalik rakendusele lisada uute andmetüüpide tuge, rakendades selleks kas uusi varasemalt arendatud teeke või luues rakendusele ise uue teegi.

Käesoleva rakenduse raames on sisendandmete andmestikku lisamine implementeeritud Discordi juturoboti abil, millele on võimalik sobilikul kujul failide saatmine. Failide eemaldamine andmestikust ning andmestiku haldamine toimub läbi Kibana keskkonna.



Joonis 1. Süsteemi arhitektuuri ülevaade.

4.2 Langfuse

Langfuse on Langfuse GmbH poolt arendatud avatud lähtekoodiga generatiivse tehisaru platvorm, mille peamiseks kasutusjuhiks on erinevate keelemudelite ning nende metaandmete analüüsimine ja kokku koondamine.

Kuigi Langfuse on paigaldatav nii eri pilvpakkujate kui ka ettevõtte enda taristusse, on projekti mahu, prioriteetidest ning eelarvest lähtuvalt otsustatud Langfuse Cloudi kasuks. See võimaldab säilitada 50 000 vaatlust igas kalendrikuus tasuta. Üheks vaatluseks loetakse ühte kasutajapoolset toimingut, antud juhul süsteemile päringu saatmist. Cloud lahendus võimaldab vähendada projekti ülalhoidmiskulusid ning hooldustööle ning ajakohastamisele kuluvaid töötunde. Lisaks aitab pilvtehnoloogiate kasutamine tagada suurema töökindluse ja kättesaadavuse. [35]

4.3 LlamaIndex

Antud projektis on skaleeritavuse ning modulaarsuse huvides otsustatud LlamaIndexi kasuks, rakendades RAG lähenemist asjakohaste allikate leidmiseks ning nende põhjal vastuste genereerimiseks. LlamaIndex pakub konteksti rikastamiseks vajalikke abstraktsioone, mis lihtsustavad andmete süsteemi lisamist, hoiustamist ning struktureerimist. [21]

Käesolevas projektis otsustati RAG süsteemide kasuks, et tagada madalamaid ülalpidamiskulusid, lihtsustada mudelimuudatusi ning suurendada mudeli toimingute läbipaistvust. Alternatiivseks lahenduseks oleks peenhäälestatud mudeli loomine olemasoleva andmestiku põhjal, kuid see suurendaks olulisel määral nii projekti kulusid, töötunde kui ka vähendaksid projekti hallatavust.

4.4 Ollama

Ollama on avatud lähtekoodiga rakendus, mis võimaldab kasutada, luua ning jagada keelemudeleid, kasutades selle haldamiseks kas MacOS / Linux käsurealiidest või Dockeri konteinerit, mis tagab eri süsteemidel toimimise ja lihtsa hallatavuse. [36]

Toetatud mudeleid on võimalik alla laadida HuggingFace'i keskkonnast ning jooksutada oma ettevõtte taristus olemasoleval riistvaral. Ollama toetab ka graafikakaardi kiirendust, mis vähendab päringutele kuluvat aega märgatavalt. Lisaks on võimalik mudeleid kiiresti katsetada ning soovi korral ka välja vahetada.

4.5 Discord

Vastavalt eelnevalt sätestatud süsteemi nõuetele, on kasutajaliidese lahenduseks valitud Discord. Discord on kasutajate seas juba varasemalt kasutuses, minimeerides uute loodavate keskkondade hulka ning optimeerides töövoogu. Lisaks on sellel ka olemasolev ekraanilugeja tugi, mis aitab muuta süsteemi ligipääsetavaks vaegnägijaile. Discordi on võimalik kasutada nii eri mobiilseadmetes kui ka arvutites ilma, et peaks suurt rõhku pöörama eri tehnoloogiate, ekraanisuuruste kui teistele tehnilistele nõuetele. [37]

Discordiga infovahetuseks on kasutusel API, mis võimaldab kasutajal nii päringut saata, sellele vastet saada, päringu väljundit hinnata kui ka end tuvastada. Üheks API võimalikuks miinuseks on striimimisfunktsionaalsuse puudumine ehk kasutajani jõuab väljund tervikuna, mitte tekstiühiku kaupa. [38]

4.6 Elastic Stack

Andmestiku hoiustamiseks vektorandmebaasis on kasutusel Elastic Stacki NoSQL andmebaas [12]. Samasugune andmebaas on TalTechis varasemalt liidestatud Moodle'i õppeanalüütika pistikprogrammiga, mis loob võimaluse soovi korral loodava süsteemi liidestamise olemasolevaga. [39]

Potentsiaalseks miinuseks Elastic Stacki kasutuses on selle hind ning jätkusuutlikkus. Tegemist ei ole vabavaralise ning avatud lähtekoodiga rakendusega, mistõttu võib tulevikus kulutuste määr tõusta. Võimalikuks alternatiiviks oleks OpenSearch, mis on Elastic Searchi vabavaraline haru [15]. Sellele vaatamata on olemasoleva andmestiku liidestamiseks antud bakalaureusetöö raames otsustatud jätkata Elastic Searchiga.

5. Arendusprotsess

Käesolev peatükk annab ülevaate süsteemi kavandamisest ning arendusprotsessist. Süvitsi tutvustatakse ka LlamaIndexi ning Langfuse'i konfiguratsioone.

5.1 Rakendatavad arendusvõtted

Rakenduse arendusprotsess põhines agiilsel arendusel. Eesmärgid olid sätestatud kahenädalaste vahetähtaegadega, kus iga vahetähtaeg omas eraldi eesmäärke ning sihte. Projekti halduseks ning arendatava rakenduse proovimiseks on kasutatud Tallinna Tehnikaülikooli GitLab keskkonda ning ka ülikooli poolt hallatavaid servereid.

Kuna antud rakendus on arendatud ainult ühe inimese poolt, siis antud projektis pole rakendatud eraldi muudatuste harude ning tõmbekutsete (ingl. k. *pull-request*) põhimõtteid.

Projektiga seonduvaks suhtluseks nii abiõppejõudude kui ka juhendajaga on kasutatud Discord platvormi, samal platvormil on testitud ka rakenduse toimimist. Juhendajaga suhtlus ning tagasisidestamine toimus jooksvalt vajadusepõhiselt.

5.2 Elastic Stack seadistused

Antud peatükk annab ülevaate Elastic Stack seadistustest ning selle komponentide kasutusest käesolevas süsteemis.

5.2.1 Elasticsearch

Vektorandmebaasi hoiustamiseks on kasutatud otsingu- ja analüüsimootorit Elasticsearch, mis põhineb vabavaralise Apache Lucene 3 tarkvaral. Juturobotile edastatavad andmestike vektorteisendid on automaatselt salvestatud ettemääratud indeksisse. Indeksiks on kollektsoon omavahel seotud JSON dokumentidest (ingl. k. *document*), igaks dokumendiks on kogum välju, kus andmed asuvad võti-väärtus paaridena, milles võti on välja nimi ning väärtus on andmetüüp, mis on kasutusel välja iseloomustuses [40]. Ühes dokumendis hoiustatakse ühe tekstiosa sisu, selle vektorteisendit ning otsinguks vajalikke metaandmeid.

Loodud lahenduses asetsevad kõik teisendid samas indeksis, sest antud töö raames

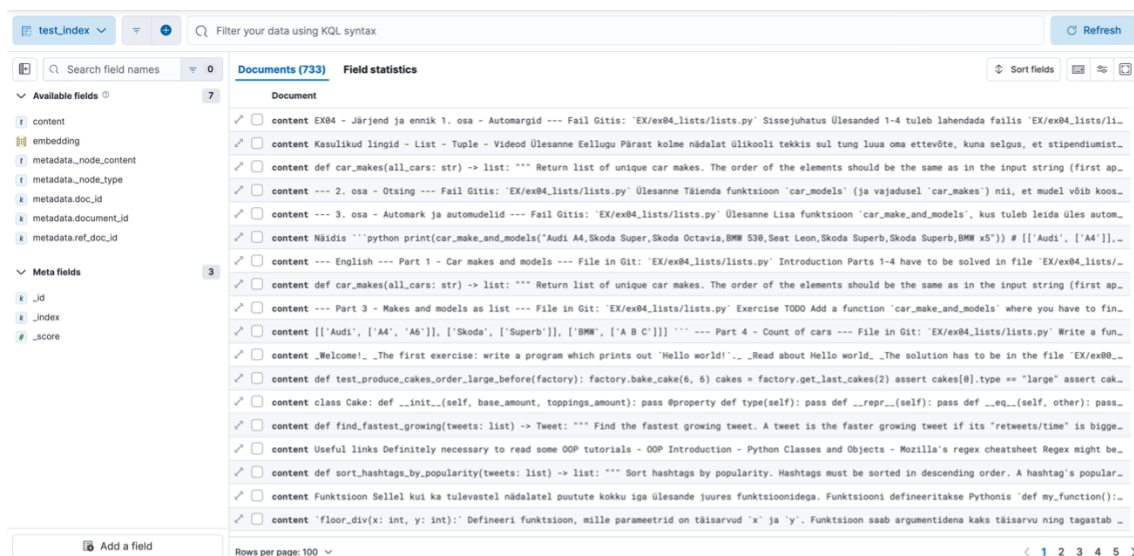
on arendatava süsteemi sihtgrupiks üks õppeaine ning tulevikus on võimalik andmete paremaks kategoriseerimiseks ning otsinguks andmestik mitme indeksi vahel ära jagada.

Elasticsearchi kiiret otsingut võimaldab inverteeritud indeksi andmetüüp, mis andmete indekseerimise käigus loob automaatselt inverteeritud indeksid, minimeerides andmepäringu protsessile kuluvat aega.

Hoiustatud andmestikku on võimalik kiirelt varundada kasutades sisseehitatud hetkvõtte (ingl. k. *snapshot*) funktsionaalsust. See võimaldab kas süsteimirikke või inimvea korral rakenduse andmestik viia tagasi eelnevatel ajahetkedel olnud kujule. [41]

5.2.2 Kibana

Kibana on käesolevas süsteemis kasutusel andmestiku haldamiseks. See on kohandatud töötamiseks automaatselt koos Elasticsearchi Dockeri konteineriga, et tagada lihtne esmane seadistus. Kibana võimaldab kasutajal vaadelda olemasolevat andmestikku ning sellega seonduvaid metaandmeid, lisaks annab see võimalust andmestikust soovi korral dokumente eemaldada. Kibana dokumendivaade on kuvatud joonisel 2.



Joonis 2. Kibana dokumendivaade.

5.3 Ollama seadistused

Eri suurte keelemudelite paigalduseks ning halduseks on kasutusel Ollama. Ollama võimaldab nii käsurea kui ka lähtekoodis tehtud muudatuste kaudu laadida HuggingFace'i varamust erinevaid vabavaraalisi keelemudeleid, mis muudab arendus- ning testimisprotsessi mugavaks ning tagab rakenduse kerge hallatavuse ning ülalpidamise pikemas perspektiivis.

Ollama toimib kui REST API server, millele on võimalik päringuid saata nii tavaliste POST sõnumite kui ka erinevate liidestuste kaudu. Käesolevas rakenduses on Ollama liidestatud kasutades LlamaIndexi Ollama integratsiooni, mis lubab tekitada ning määrata vastavad parameetrid otse Pythoniga rakenduse lähtekoodis. [42]

Suurte keelemudelite ressursinõudlusest tulenevalt on oluline keelemudeli valikul lähtuda süsteemi arhitektuurist ning komponentidest, täpsemalt graafikakaardi (GPU) ning muutmälu (RAM) olemasolust ning kasutusvõimalusest. Arendustöö jooksul on Ollamat kasutatud nii lokaalselt MacOS operatsioonisüsteemiga Apple M1 Pro protsessoriga arvutil, millel on 16GB muutmälu, kui ka läbi Dockeri vahekihi, mis isoleerib Ollama instantsi Docker konteinerisse. Kuigi mõlemal juhul on võimaldatud ressursside hulk sama, võimaldab lokaalselt MacOSi paigaldatud Ollama kasutada graafikakaardipõhist kiirendust, muutes vastusegenereerimisprotsessi kordades kiiremaks ning efektiivsemaks. Sarnaselt MacOS süsteemidega, toetab Ollama Linux masinatel graafikakaardipõhist kiirendust ning paigalduse jooksul installitakse ka vajalikud AMD Radeon ja Nvidia CUDA graafikakaardi draiverid. [43]

Sellest lähtudes on arendatud rakenduses jäetud kasutajale võimalus kasutada Ollamat nii läbi Docker konteineri kui ka lokaalselt jooksutades, rakendades kohaliku masina graafikakaarti protsessi kiirendamiseks.

5.4 Rakenduse paigaldus ning käivitus

Rakendus on terviklikult paigaldatav Docker konteinerite kujul. Valmistatud on paigaldust lihtsustav *docker-compose* fail, mis võimaldab väheste sammudega rakendust paigaldada. See tekitab paigaldavasse masinasse uue Docker võrgu, et tagada sujuv konteineritevaheline infovahetus, Kibana, Elasticsearchi konteinerid andmete halduseks ning hoiustamiseks, LlamaIndex rakenduse konteineri päringute halduseks ja Discordiga suhtluseks ning ka vajadusel Ollama konteineri suurte keelemudelite halduseks. Ollama konteineri asemel on võimalik seda ka paigaldada otse masinale, võimaldades seeläbi graafikakaardipoolse kiirenduse kasutust.

Lisaks eelnevalt mainitule, on vajalik Discordi rakenduse seadistus vastavast arendaja kasutajaliidesest ning lisada seal genereeritud vastav API võti keskkonnamuutujate sekka.

Langfuse'i kasutuseks on vaja kas luua kasutaja Langfuse Cloud keskkonda või paigaldada vajalikud Langfuse'i konteinerid samuti lokaalsele masinale ning valitud Langfuse'i instantsist luua API võti ja see keskkonnamuutujate sekka lisada.

5.5 Langfuse

Rakenduse ning logide lihtsaks hallatavuseks on kasutusel Langfuse, mis koondab kokku kõik päringutega kaasnevad metaandmed, kasutusstatistika, vastuste kvaliteediga seonduva meetrika ning ka mudeli vastuseid mõjutavad viibad.

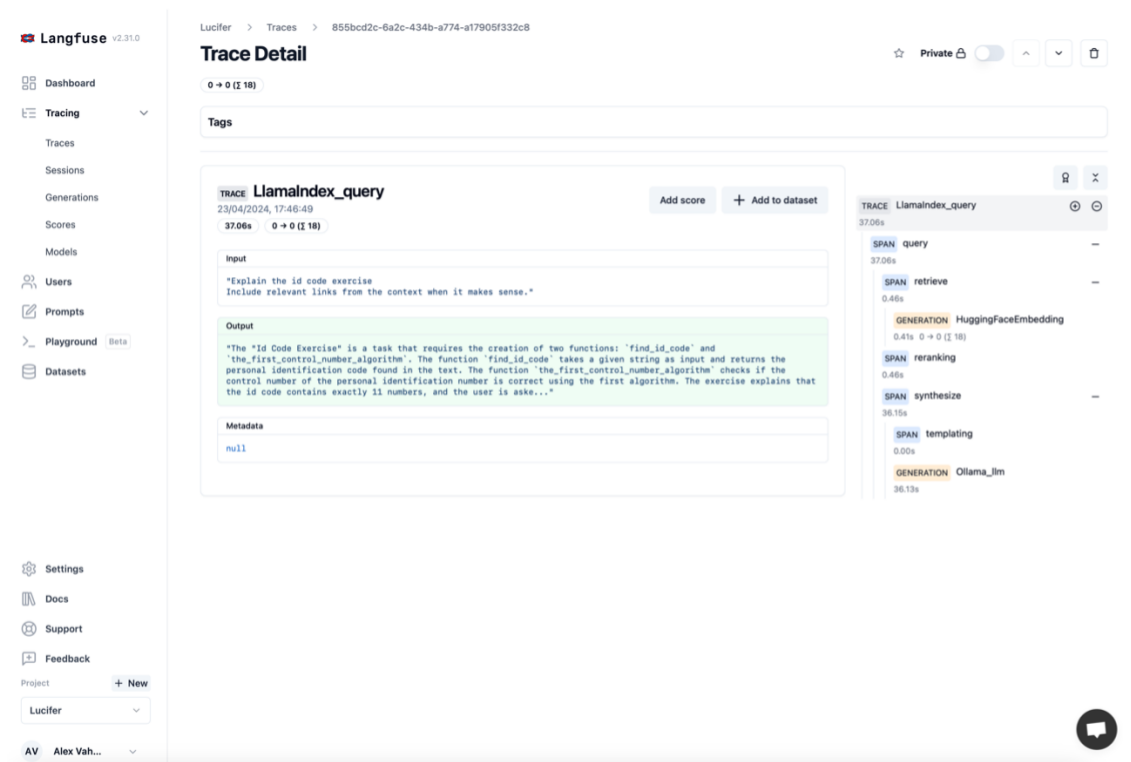
Käesolevas rakenduses on Langfuse'i instants seadistatud Langfuse Cloud keskkonda, mis tagab keskkonna kõrge stabiilsuse, madalad ülalpidamiskulud ning lihtsa kättesaadavuse eri kasutajatele. Võimalik on ka Langfuse paigaldada lokaalselt Docker konteineritesse, kuid Langfuse'i kiire arendusprotsessiga nõuaks selline lähenemine sagedast manuaalset uuendusprotsessi. Langfuse on liidestatud kasutades Langfuse'i Llamaindexi liidest.

5.5.1 Päringute logimine

Kõik rakenduse poole pöörduvad päringud on salvestatud Langfuse'i keskkonda vaatlustena. Üks vaatlus on antud kasutusjuhul üks kasutajapoolne päring ehk küsimus või sõnum juturobotile.

Langfuse'i LlamaIndex liides tagab vajaliku logimise lõppkasutaja- või arendajapoolse lisatööta ehk kogu päringuga kaasnev andmestik on automaatselt kategoriseeritud loometeks (ingl. k. *generation*) ja kestvusteks (ingl. k. *span*), mis aitab muuta taustal toimuvad protsessid läbipaistvamaks ning tagab vajadusel lihtsa ja loogilise probleemiotsingu.

Iga vaatluse alatoiming hõlmab selle toiminguga jaoks vajalikku infot, loomed on kasutusel uute andmete genereerimisel, olgu selleks kas vektorteisendite või päringute vastuste loomel. Kestvused on kasutusel toimingute ajalise kulu ning sisendite ja väljundite talletamiseks. Joonis 3 kujutab Langfuse'i vaatluse sisu.



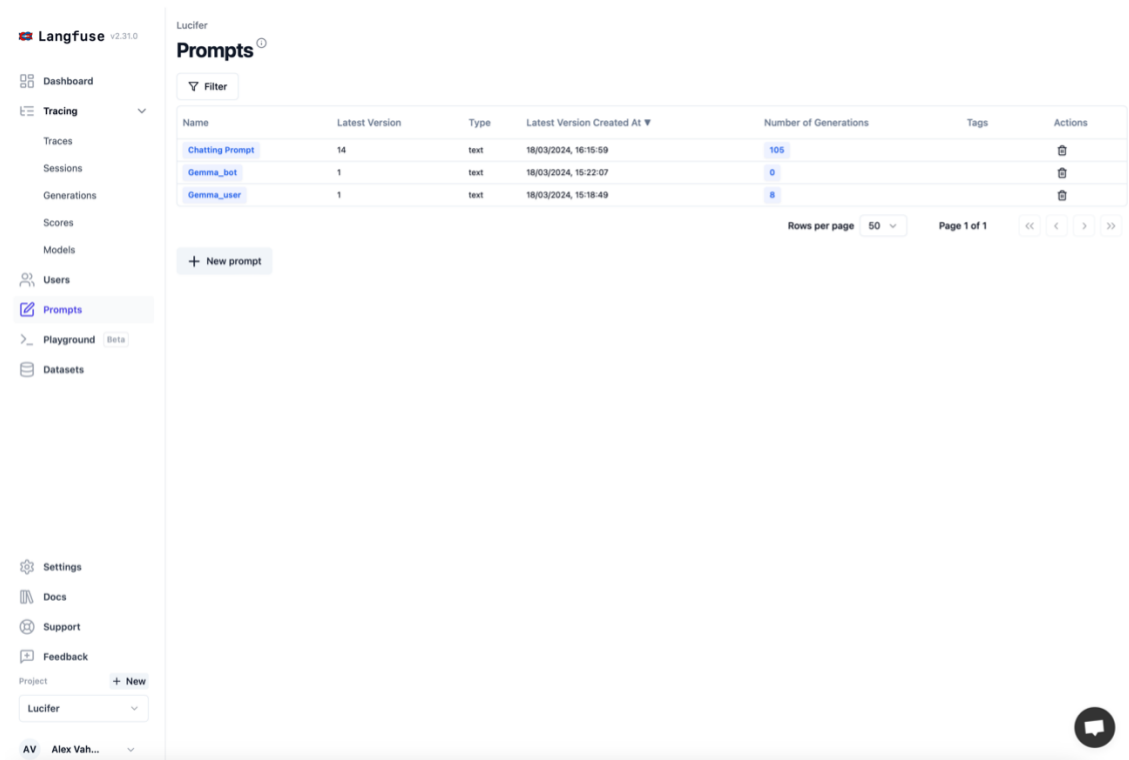
Joonis 3. Langfuse'i *Trace Detail* vaade.

5.5.2 Viibahaldus

Lisaks päringus sisalduva kasutajajuhisele, on võimalik vastuse mõjutamiseks kaasata viipa, mille abil saab suunata vastuse loomet soovitud kuju poole.

Selleks, et viipasid jooksvalt kohandada ning muuta, on kasutusel Langfuse'i Prompts keskkond, mis tagab ajakohaste viipade edastamise tagarakendusse. Lisaks on viipade muutmisajalugu salvestatud ning seostatud varasemalt sooritatud päringutega, mis aitab võrrelda vastuste kvaliteeti ning selle sõltuvust viipadest.

Hoiustades viipasid eraldi keskkonnas ning vajadusepõhiselt pärides on võimalik viipasid muuta rakendust uuesti ehitamata ning paigaldamata, see lihtsustab mudelite vahetuse ning vastuse kvaliteedi parendamise protsesse märgatavalt. Langfuse'i viipade vaade on kuvatud joonisel 4.



Name	Latest Version	Type	Latest Version Created At	Number of Generations	Tags	Actions
Chatting Prompt	14	text	18/03/2024, 16:15:59	105		
Gemma_bot	1	text	18/03/2024, 15:22:07	0		
Gemma_user	1	text	18/03/2024, 15:18:49	8		

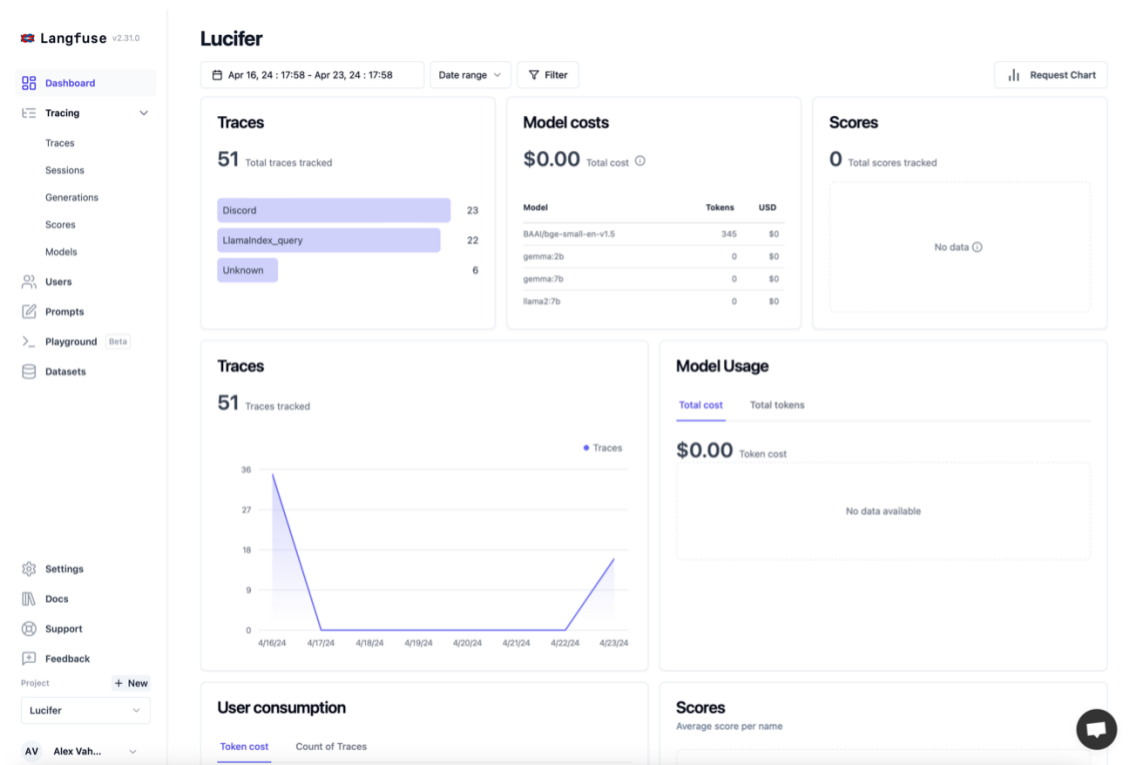
Rows per page: 50 | Page 1 of 1

+ New prompt

Joonis 4. Langfuse'i *Prompts* vaade.

5.5.3 Statistika

Toimingute jälgimise lihtsustamiseks on võimalik vaadelda Langfuse'i töölauda, kuvatud joonisel 5, mis annab lihtsustatud ülevaate päringute arvust, vastuste genereerimisele kuluvast ajast ning ka tasuliste mudelite ning APIde kasutuse hinnastusest. Nimelt arvutab Langfuse automaatselt sisendi ja väljundi tekstiühikute arvu ning nendele põhinedes ka kulusumma. Antud rakendus on paigaldatud lokaalselt toimimaks kohalike mudelitega, mistõttu pole kulusumma väli väärtustatud.



Joonis 5. Langfuse'i Dashboards vaade.

5.6 Sobiva suure keelemudeli valik

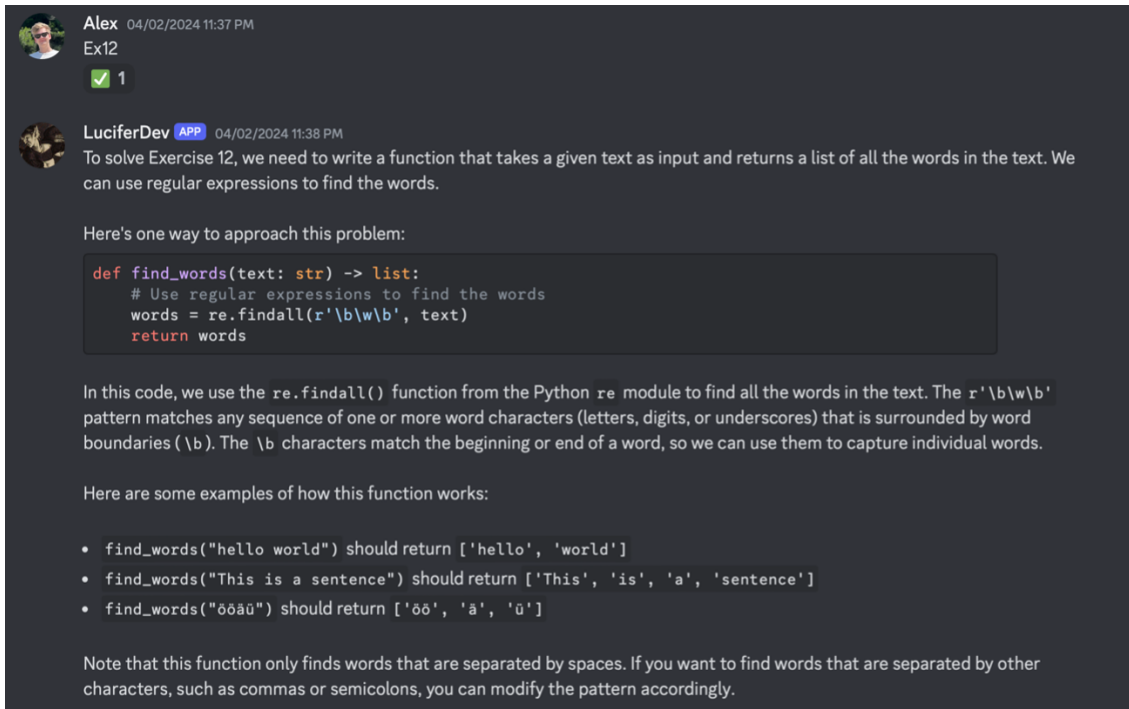
Mudelivalikul on sätestatud kaks peamist eesmärki: mudeli kättesaadavus ning ressursside kasutuse ning selle vastuse kvaliteedi vaherkord. Antud peatükk annab ülevaate kaalutud mudelitest ning nende tulemustest.

5.6.1 Gemma

Google Gemma on Google'i poolt arendatud ning treenitud ressursisäästlik mudel, mis on ehitatud samadele põhimõtetele toetudes nagu Google Gemini. Kuna tegemist on võrdlemisi väiksemahulise keelemudelig, on seda võimalik kasutada paljudel erinevatel seadmetel, aidates vähendada seeläbi ka projekti ressursinõudeid ning nendest tulenevaid piiranguid. Mudel on vabalt kättesaadav ning ei vaja eraldi Google'ile taotluse esitamist ega heakskiitu, siiski on vajalik sätestatud kasutustingimustega nõustumine. [44]

Käesolevas projektis on kasutusel Gemma suurte keelemudelite võimsaim mudel Gemma-7b, mis kasutab 7 miljardit parameetrit oma vastuste genereerimisel. Kuigi tegemist on suurima Gemma mudeliga, on see siiski eelkõige ingliskeelsetel materjalidel treenitud, mistõttu on rakendus kasutatav vaid ingliskeelse sisuga andmete ja käsklustega. Gemma

mudelite suureks eeliseks on koodi mõistmine ning väljastamine *Markdown* kujul, mis lihtsustab infopäringut koodilõikusi sisaldavatest failidest. Selline väljund on kuvatud joonisel 6.



Alex 04/02/2024 11:37 PM
Ex12
✓ 1

LuciferDev [APP] 04/02/2024 11:38 PM
To solve Exercise 12, we need to write a function that takes a given text as input and returns a list of all the words in the text. We can use regular expressions to find the words.

Here's one way to approach this problem:

```
def find_words(text: str) -> list:  
    # Use regular expressions to find the words  
    words = re.findall(r'\b\w\b', text)  
    return words
```

In this code, we use the `re.findall()` function from the Python `re` module to find all the words in the text. The `r'\b\w\b'` pattern matches any sequence of one or more word characters (letters, digits, or underscores) that is surrounded by word boundaries (`\b`). The `\b` characters match the beginning or end of a word, so we can use them to capture individual words.

Here are some examples of how this function works:

- `find_words("hello world")` should return `['hello', 'world']`
- `find_words("This is a sentence")` should return `['This', 'is', 'a', 'sentence']`
- `find_words("ööü")` should return `['öö', 'ä', 'ü']`

Note that this function only finds words that are separated by spaces. If you want to find words that are separated by other characters, such as commas or semicolons, you can modify the pattern accordingly.

Joonis 6. Juturoboti koodilõigu genereerimine.

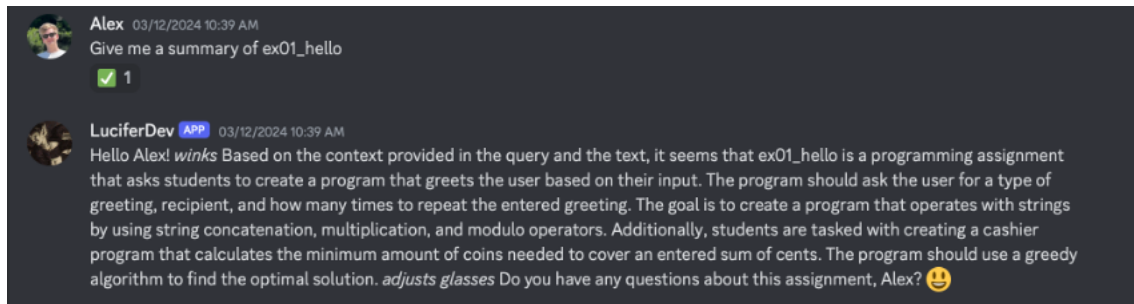
Lisaks on 7 miljardit parameetrit suurte keelemudelite kontekstis võrdlemisi väike arv, mistõttu peab arvestama madalama vastusekvaliteediga ja ebastabiilsema vastuste sisuga. Väikse mudeli suureks eeliseks on küll vastuse kiirem genereerimine, mis pakub kasutajale parema kasutuskogemuse, eriti antud rakenduse kontekstis, kus kasutajaliides, Discord, ei võimalda vastuste striimimist.

5.6.2 Llama2

Llama2 on Meta poolt arendatud vabavaraline keelemudel, millest leidub 7, 13 kui ka 70 miljardi parameetriga variante. Parameetrite hulga suurenemisega paraneb vastuste kvaliteet, sest mudel on võimeline kontekstist paremini infot eraldama. Miinusküljeks on aga süsteemi suurem ressursinõudlus ning kiiruse vähenemine. See tähendab, et suureneb nii energia- kui ka ülalpidamiskulu. Taaskord on tegu eelkõige ingliskeelse mudeliga, mistõttu esineb probleeme eestikeelse andmestiku kasutamisega. [45]

Antud rakenduse kontekstis on proovitud nii 7 kui ka 13 miljardise parameetriga Llama2 variante ning vastusekvaliteedi vahe ei olnud suuresti märgatav, mistõttu osutus vastuse

genereerimiskiirus määravaks teguriks ning kasutusse läks 7-miljardilise parameetriga mudel. Mõlemal testitud Llama2 mudelil esines kohati probleeme vastuse sõnastusega, kus vastus esitati vabas vormis, kasutades nii kaldkirjas emotsioonide ja tegevuste väljendamist kui ka *emoji*'sid. See probleem on kujutatud joonisel 7 ning esines näiliselt suvalistel ajahetkedel rakenduse lähtekoodi muutmata, mistõttu on mudeli usaldatavus küsimärgi all. Siiski tundus vastuste kvaliteet ühtlaselt parem kui Gemma mudelite oma.



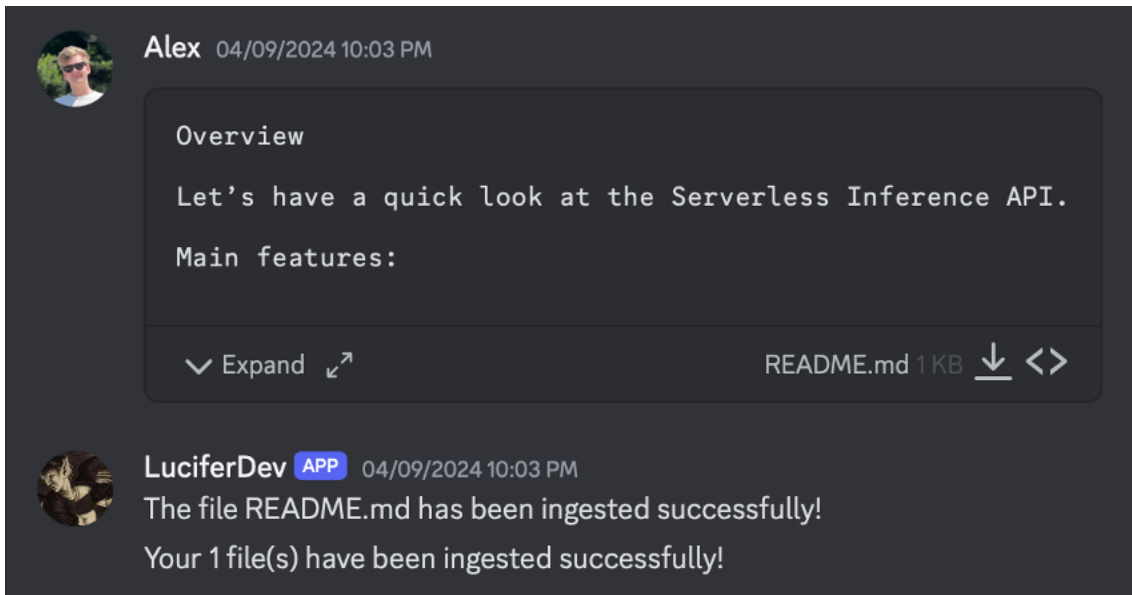
Joonis 7. Juturoboti vastusekvaliteedi näide.

5.7 Discord

Loodud süsteemi kasutajaliidesena on valitud Discordi platvorm, mis on juba varasemalt laialdaselt kasutusel Tallinna Tehnikaülikooli õppetöö korraldamises. Rakenduse liidestamine olemasoleva süsteemiga lihtsustab uue süsteemi kasutuselevõttu ning vähendab aega, mis kulub süsteemi halduseks ning hooldamiseks. Discord võimaldab ka rakendusel toimida suure hulga eri seadmetega, mistõttu on rakendus laialdaselt kättesaadav ning käepärane. Discord on optimeeritud ka ekraanilugeritega toimimiseks, muutes rakendust kättesaadavaks ka vaegnägijatele.

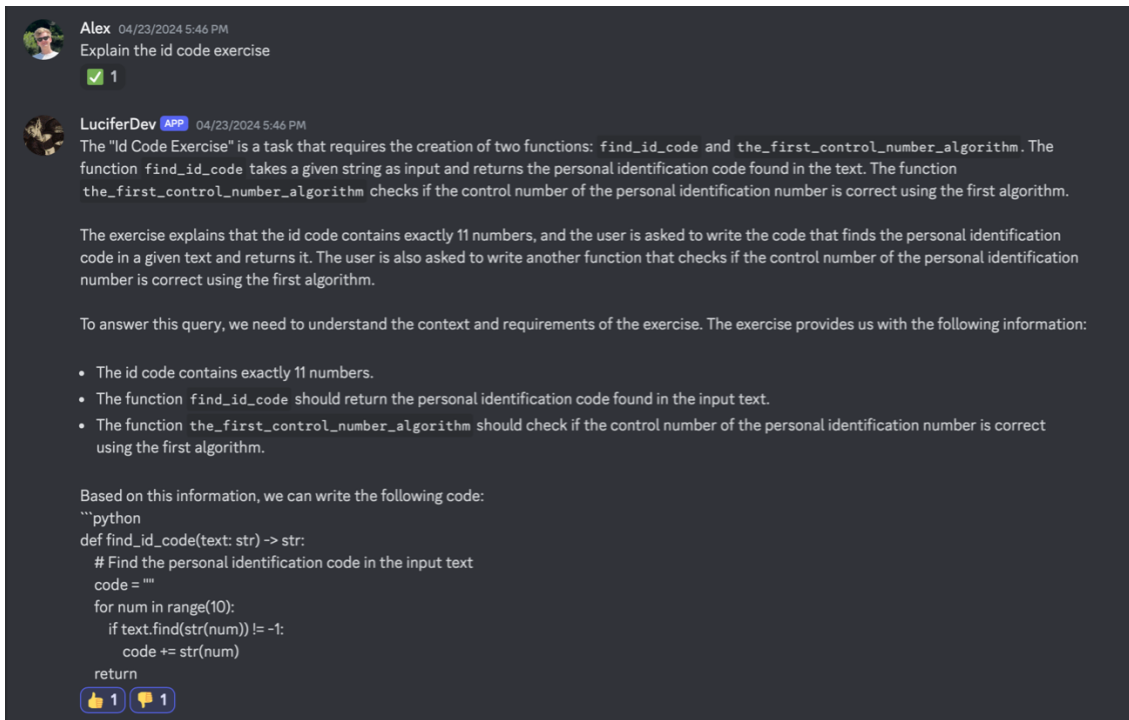
Rakenduse kasutamiseks on vajalik juturoboti paigaldus soovitasse Discordi serverisse ning kas Discordi kanalis või privaatsõnumites juturobotiga suhtlemine. Juturobotiga suhtluse on võimalik jagada alamosadeks: info edastamine juturobotile, info tagasisidestamine ning info pärimine juturobotilt.

Selleks, et oleks võimalik infot pärida, on vajalik selle eelnevalt edastamine juturobotile. See on implementeeritud failide manustamisena. Nimelt on kasutajal vajalik vaid juturobotile failid manusena saata, käivitades sellega automaatselt failide vektorteisendite loomise ning tagarakenduses andmebaasi talletuse. Kasutaja saab vastava sõnumi protsessi õnnestumisest. See protsess on kujutatud joonisel 8.



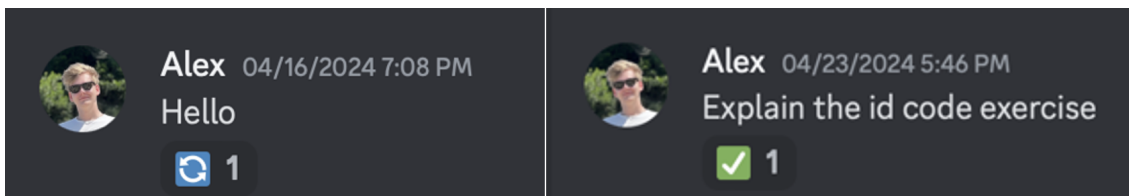
Joonis 8. Juturoboti sisendandmete laadimine.

Info tagasisidestamine toimub sõnumitele pöial-üles ja pöial-alla reaktsioonidega. Tagasisidestusprotsess on kuvatud joonisel 9. Saadetud tagasiside seostatakse ning talletatakse automaatselt tagasisidestatud sõnumiga ning on vaadeldav Langfuse keskkonnas vaatluste alt. Juhul kui juturobotiga suhtlus toimub Discordi kanalis ning üht sõnumit tagasisidestab mitu kasutajat, salvestatakse kõigi kasutajate tagasisided ning Langfuse'is on võimalik näha iga kasutaja personaalset hinnangut. Saadud hinnanguid on võimalik vaadelda ka keskmise hinnangu kujul. Seeläbi on võimalik eristada kasutajate personaalseid eelistusi vastuste kuju ning sõnastuse osas.



Joonis 9. Juturoboti vastuse tagasisidestamine.

Päringutele vastuse genereerimise indikaatorina on kasutuses kasutaja sõnumile reageerimine, kuvatud joonisel 10, kus juturobot märgistab vastust ootava päringu sõnumi nooltega *emoji*’ga ning vastuse genereerimise õnnestumisel ning selle eduka edastamisega vahetatakse noolte kujutis rohelise linnukesega. See võimaldab kasutajal saada kinnitust, et tema päring on jõudnud kohale. Päringute staatuse tagasisidestamine on eriti oluline Discordi sõnumite kontekstis, sest Discord ei võimalda vastuste reaajas edastamist, mistõttu on vajalik vastuse terviklik genereerimine enne vastuse kasutajale edastamist. Seda heastab fakt, et vastuste genereerimine on taustprotsess, mis edastab valminud vastuse sõnumina sõltumata sellest, kas kasutaja lahkub Discordist või mitte.



Joonis 10. Juturoboti vastuse genereerimise staatus.

6. Tulemused

Antud peatükis antakse ülevaade käesoleva bakalaureusetöö tulemustest ning saavutustest. Esmalt käsitletakse loodud funktsionaalsuseid ning seejärel vaadeldakse süsteemi edasiarenduse võimalusi.

6.1 Loodud funktsionaalsused

Töö raames saavutati enamus sätestatud eesmärkidest ning loodi õppeinfot analüüsiv ning küsimustele vastav süsteem. Loodud süsteem koosneb kuuest põhikomponendist, milleks on Discord, LlamaIndex, Ollama, Langfuse, Elasticsearch ja Kibana. See võimaldab kasutajal kiirelt ning mugavalt sisendandmete kohta küsimusi küsida vabamas vormis ning analüüsida mitmest erinevast andmestikust infot. Eraldi pöörati rõhku õppeinfole nagu Moodle'i ülesanded, kuid süsteemi ülesehitusest lähtuvalt on süsteem laiendatav ka teistele andmeallikatele. Erinevalt varasemalt kasutusel olevatest vahenditest ei vaja loodud lahendus eelnevaid teadmisi ning lubab süsteemiga suhelda vabas vormis.

Süsteemi kasutamiseks on vajalik tagarakenduse paigaldus soovitavasse masinasse ning tagarakenduse liidestamine Discordi arendajaportaalis loodud Discordi juturoboti instantsiga. Antud liidestus loob kasutajale keskkonna, kus rakendusega suhelda ning faile ning päringuid edastada. Lisaks on vajalik Langfuse Cloud konto loome ning selle API võtme edastamine rakendusele, mis võimaldab viipade, logide ja metaandmete talletust ja haldust.

6.2 Edasiarenduse võimalused

Käesoleva bakalaureusetöö raames loodud süsteemi on võimalik mitmekülgsest edasi arendada. Hetkel realiseeriti päringutele vastuse genereerimine lähtuvalt sisendandmetest Discordi juturoboti abil, kuid süsteemi on võimalik liidestada ka teiste keskkondadega nagu Microsoft Teams või Facebook Messenger, et pakkuda kasutajatele esirakenduse osas valikut ning vähendada veelgi uute rakenduste õppimise vajadust. Soovi korral on võimalik luua ka eraldi veebileht esirakenduseks, see võimaldaks suuremat kontrolli nii kasutajakogemuse kui ka edastatava info üle.

Esialgsest oli plaanis automaatse andmete laadimise võimaldamine ning liidestamine olemasoleva Moodle'i pistikprogrammi ning selle tagarakenduse Elastic Stackiga.

Süsteemi liidestamiseks oleks vajalik vektorteisendite genereerimine Moodle'i pistikprogrammi Elasticsearch instantsis ning selle vormingu ühtsustamine käesoleva rakenduse Elasticsearchi ülesehitusega. Kuigi käesoleva rakenduse arenduse kavandamisel oli see integratsioon planeeritud ning sellest lähtuvalt valiti Elastic Stack andmestiku hoiustamiseks ning halduseks, jäi ajapuuduse ning suure mahu tõttu kahe süsteemi liidestamine ära.

Lisaks eelnevalt mainitule oleks võimalik käesolevale rakendusele lisada ka rohkemate andmetüüpide ning -allikate tugi. See võimaldaks rohkemate andmete kasutust ning seeläbi ka põhjalikumaid ning intuitiivsemaid päringuid ja analüüse. Süsteem on arhitektuurilt modulaarse ülesehitusega ja kasutab LlamaIndexi integratsioone ning mooduleid eri keskkondade ning andmetüüpide toetamiseks, mistõttu on süsteem üsna lihtsasti vajadusepõhiselt laiendatav ning kohandatav.

Võimalik oleks lisada ka andmestiku ning sellel põhinevate vastuste ja päringute automaattestimine ning tagasisidestamine. Antud süsteemi liidestatud Langfuse'i keskkonnal leidub tugi erinevate meetrikate arvutuseks, mis loob parema ülevaate vastuste muutumisest ajas ning toob esile võimalikud kasutusjuhud, millega võib juturobotil probleeme esineda.

Süsteem on ülesehituselt skaleeritav, mistõttu on võimalik ressursside hulka suurendades muuta rakendus kättesaadavaks suurele kasutajaskonnale. Sellisel kasutusjuhul oleks vajalik põhjalikum kasutaja- ning andmehalduse süsteem. Hetkel on kasutusel kasutajate halduseks Discordi juturobotipoolsed turvameetmed, mis võimaldavad rakenduse kättesaadavust limiteerida juturoboti paigaldusasukohast, kanalist või kasutajast lähtuvalt. Sellele vaatamata võib esineda probleem mitme esirakenduse kasutusega, mil korraga leidub kasutajaid erinevatel platvormidel, mis vajavad kõik eraldi filtrite ning piirangute loomet. Seda olukorda oleks võimalik parendada ühtse kasutajasüsteemi abil, mis muudaks ka kasutajate halduse tsentraalsemaks.

Antud süsteem kasutab päringute ning vektorteisendite genereerimiseks lokaalseid mudeleid. Vastuse kvaliteedi, kiiruse, skaleeritavuse ning töökindluse huvides oleks võimalik Ollama asemel kasutada ka veebipõhiseid suuri keelemudeleid erinevatelt pakkujatelt nagu OpenAI. Lisaks oleks võimalik rakendus paigaldada tervikuna pilvteenusepakkuja juurde. Selliste otsuste langetamine vajab aga põhjalikumalt analüüsi nii hinnastuse, jätkusuutlikuse kui ka kasutusjuhtude osas.

7. Kokkuvõte

Käesoleva bakalaureuse töö eesmärgiks oli luua Tallinna Tehnikaülikoolile pilootrakendus, mille abil oleks võimalik lihtsalt ja intuiitselt õppeinfo seotuid andmeid pärida ning analüüsida. Lisaeesmärkideks oli minimeerida uute keskkonda loomist ning tagada süsteemi toimimine ka ekraanilugejatega. Kõik eelnevalt mainitu ka saavutati. Loodud süsteemi miinuseks on aga automaatne andmete uuendamine ning pärimine erinevatest õppeinfo keskkondadest. Sellele liidestusele on loodud arendustöö jooksul esialgne tugi, kuid toimiv funktsionaalsus ning implementatsioon puudub.

Lõputöö eesmärgi saavutamiseks sätestati piiratud skoop ning viidi läbi kasutusel olevate keskkondade analüüs ning käsitleti planeeritava süsteemi võimalikke alternatiivkomponente ning nende mõju. Analüüsi tagajärjena defineeriti funktsionaalsed ning mittefunktsionaalsed nõuded ning sellele põhinedes loodi projekti arhitektuur. Käesolev süsteem on arendatud Python programmeerimiskeeles ning koosneb kuuest komponendist: Discord, LlamaIndex, Ollama, Langfuse, Elasticsearch ja Kibana.

Antud süsteemil leidub ka mitmeid erinevaid edasiarenduse suundi ning võimalusi. Üheks oleks olemasolevate õppekeskkondade liidestamine loodud süsteemi, mille abil oleks võimalik reaalajas andmeid importida ning uuendada, tagades seejärel, et kõik päringud kasutaksid vaid kõige uuemat andmestikku. Lisaks oleks võimalik suurendada süsteemis toetatud andmetüüpide hulka, võimaldades seeläbi rohkemate päringute ning andmete põhjal vastamist. Süsteemi edasi arendamisel on võimalik pakkuda terviklikumat süsteemi, mis võimaldab veelgi täpsemalt ning kiiremalt päringutele vastata ning võimalikke õppeinfo murekohti tuvastada.

Kasutatud kirjandus

- [1] Salesforce. *12 Benefits of Cloud Computing*. 2024. URL: <https://www.salesforce.com/products/platform/best-practices/benefits-of-cloud-computing/> [Kasutatud: 9.5.2024].
- [2] A. Larkin. *Disadvantages of Cloud Computing*. July 2023. URL: <https://cloudacademy.com/blog/disadvantages-of-cloud-computing/> [Kasutatud: 9.5.2024].
- [3] Oxford. *On-Premise, Cloud, or Both? Four Considerations to Build Your Strategy*. Sept. 2020. URL: <https://www.oxfordcorp.com/en/on-premise-cloud-or-both-what-you-should-know-before-deciding/> [Kasutatud: 10.5.2024].
- [4] Microsoft. *Cloud economics*. URL: <https://azure.microsoft.com/en-us/solutions/cloud-economics/> [Kasutatud: 10.5.2024].
- [5] Microsoft. *Data import and analysis options*. URL: <https://support.microsoft.com/en-us/office/data-import-and-analysis-options-3ea52160-08bc-45ac-acd9-bc4a11bcc2a2> [Kasutatud: 10.5.2024].
- [6] Microsoft. *Microsoft Excel*. URL: <https://www.microsoft.com/et-ee/microsoft-365/excel> [Kasutatud: 10.5.2024].
- [7] D. Raymond. *Top 10 Cons and Disadvantages of Using Microsoft Excel*. Dec. 2023. URL: <https://projectmanagers.net/top-10-cons-disadvantages-of-using-microsoft-excel/> [Kasutatud: 10.5.2024].
- [8] Google. *Make data-driven decisions, in Google Sheets*. URL: <https://www.google.com/sheets/about/> [Kasutatud: 10.5.2024].
- [9] D. Raymond. *Google Sheets: Top 10 Cons and Disadvantages*. Dec. 2023. URL: <https://projectmanagers.net/google-sheets-top-10-cons-and-disadvantages/> [Kasutatud: 10.5.2024].
- [10] Discord. *WHAT IS DISCORD?* May 2022. URL: <https://discord.com/safety/360044149331-what-is-discord> [Kasutatud: 10.5.2024].
- [11] Microsoft. *Copilot for Microsoft 365*. URL: <https://adoption.microsoft.com/en-us/copilot/> [Kasutatud: 10.5.2024].

- [12] Elasticsearch. *Meet the search platform that helps you search, solve, and succeed.* 2024. URL: <https://www.elastic.co/elastic-stack> [Kasutatud: 23.4.2024].
- [13] Elasticsearch. *The heart of the free and open Elastic Stack.* URL: <https://www.elastic.co/elasticsearch> [Kasutatud: 10.5.2024].
- [14] Elasticsearch. *Elastic pricing.* URL: <https://www.elastic.co/pricing> [Kasutatud: 10.5.2024].
- [15] OpenSearch. *About OpenSearch.* Dec. 2022. URL: <https://opensearch.org/about.html> [Kasutatud: 10.5.2024].
- [16] OpenSearch. *About OpenSearch.* Dec. 2022. URL: <https://opensearch.org/about.html> [Kasutatud: 23.4.2024].
- [17] D. K. Jain. *Setup OpenSearch multi-node cluster on Kubernetes using Helm Charts.* Nov. 2021. URL: <https://opensearch.org/blog/setup-multinode-cluster-kubernetes/> [Kasutatud: 10.5.2024].
- [18] Deepset. *What is Haystack?* URL: <https://haystack.deepset.ai/overview/intro> [Kasutatud: 10.5.2024].
- [19] Deepset. *Introduction to Integrations.* URL: <https://docs.haystack.deepset.ai/docs/integrations> [Kasutatud: 10.5.2024].
- [20] Deepset. *Introduction to Haystack 2.0.* URL: <https://docs.haystack.deepset.ai/docs/intro> [Kasutatud: 10.5.2024].
- [21] LlamaIndex. *Turn your enterprise data into production-ready LLM applications.* 2024. URL: <https://www.llamaindex.ai/> [Kasutatud: 23.4.2024].
- [22] LlamaIndex. *Turn your enterprise data into production-ready LLM applications.* URL: <https://www.llamaindex.ai> [Kasutatud: 10.5.2024].
- [23] LangChain. *The largest community building the future of LLM apps.* URL: <https://www.langchain.com/langchain> [Kasutatud: 10.5.2024].
- [24] LangChain. *Get your LLM app from prototype to production.* URL: <https://www.langchain.com/langsmith> [Kasutatud: 10.5.2024].
- [25] LangChain. *Turn your LLM app into an API server instantly with LangServe.* URL: <https://www.langchain.com/langserve> [Kasutatud: 10.5.2024].
- [26] LangFuse. *Open Source LLM Engineering Platform.* URL: <https://langfuse.com> [Kasutatud: 23.4.2024].
- [27] Langfuse. *Prompt Management.* URL: <https://langfuse.com/docs/prompts/get-started> [Kasutatud: 10.5.2024].

- [28] Langfuse. *Evaluation of RAG pipelines with Ragas*. URL: https://langfuse.com/guides/cookbook/evaluation_of_rag_with_ragas [Kasutatud: 10.5.2024].
- [29] R. Merritt. *What Is Retrieval-Augmented Generation, aka RAG?* Nov. 2023. URL: <https://blogs.nvidia.com/blog/what-is-retrieval-augmented-generation/> [Kasutatud: 23.4.2024].
- [30] LangChain. *ChatGPT Over Your Data*. Feb. 2023. URL: <https://blog.langchain.dev/tutorial-chatgpt-over-your-data/> [Kasutatud: 23.4.2024].
- [31] AWS. *What is GPT? 2024*. URL: <https://aws.amazon.com/what-is/gpt/> [Kasutatud: 23.4.2024].
- [32] L. Craig. *Fine-tuning*. Feb. 2024. URL: <https://www.techtarget.com/searchenterpriseai/definition/fine-tuning#:~:text=Fine%2Dtuning%20is%20the%20process,suit%20more%20specialized%20use%20cases> [Kasutatud: 23.4.2024].
- [33] OpenAI. *Fine-tuning*. 2024. URL: <https://platform.openai.com/docs/guides/fine-tuning> [Kasutatud: 23.4.2024].
- [34] LlamaIndex. *SimpleDirectoryReader*. URL: https://docs.llamaindex.ai/en/stable/module_guides/loading/simpledirectoryreader/ [Kasutatud: 10.5.2024].
- [35] Langfuse. *Open Source LLM Engineering Platform*. URL: <https://langfuse.com> [Kasutatud: 10.5.2024].
- [36] Ollama. *Get up and running with large language models*. 2024. URL: <https://ollama.com> [Kasutatud: 23.4.2024].
- [37] Discord. *Activity Design Patterns*. URL: <https://discord.com/developers/docs/activities/design-patterns#accessibility-over-exclusion> [Kasutatud: 23.4.2024].
- [38] Discord. *Community Resources*. URL: <https://discord.com/developers/docs/topics/community-resources#interactions> [Kasutatud: 23.4.2024].
- [39] A. Riisimäe and K.-M. Veinberg. *Moodle'i õpitulemuste analüüsimiseks kasutatava pistikprogrammi ja Elastic Stacki süsteemi arendamine*. A development of a plugin for analyzing Moodle learning outcomes using the Elastic Stack system. Tallinna Tehnikaülikool. 2023.

- [40] Elasticsearch. *Data in: documents and indices*. URL: <https://www.elastic.co/guide/en/elasticsearch/reference/current/documents-indices.html> [Kasutatud: 10.5.2024].
- [41] Elasticsearch. *Snapshot and restore*. URL: <https://www.elastic.co/guide/en/elasticsearch/reference/current/snapshot-restore.html> [Kasutatud: 10.5.2024].
- [42] jmorganca. *API*. May 2024. URL: <https://github.com/ollama/ollama/blob/main/docs/api.md> [Kasutatud: 10.5.2024].
- [43] moresearch. *Ollama on Linux*. May 2024. URL: <https://github.com/ollama/ollama/blob/main/docs/linux.md> [Kasutatud: 10.5.2024].
- [44] Google. *Gemma Open Models*. URL: <https://ai.google.dev/gemma> [Kasutatud: 10.5.2024].
- [45] Meta. *Llama 2: open source, free for research and commercial use*. URL: <https://llama.meta.com/llama2/> [Kasutatud: 10.5.2024].

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Alex Vahter

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose “Suurtel keelemudelitel põhineva õppeinfo päringute ja analüüsi süsteemi arendamine”, mille juhendaja on Ago Luberg
 - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

25.05.2024

¹Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingu tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtjaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

Lisa 2 – Langfuse Trace’i sisu

The screenshot displays the Langfuse Trace Detail page for a specific trace. The interface includes a sidebar on the left with navigation options like Dashboard, Tracing, Sessions, Generations, Scores, Models, Users, Prompts, Playground (Beta), and Datasets. The main content area is titled 'Trace Detail' and shows the following information:

- Trace ID:** 855bcd2c-6a2c-434b-a774-a1790f332c8
- Trace Name:** LlamaIndex_query
- Timestamp:** 23/04/2024, 17:46:49
- Duration:** 37.06s
- Score:** 0 → 0 (18)

The trace details are organized into sections:

- Input:** "Explain the id code exercise. Include relevant links from the context when it makes sense."
- Output:** "The 'Id Code Exercise' is a task that requires the creation of two functions: 'find_id_code' and 'the_first_control_number_algorithm'. The function 'find_id_code' takes a given string as input and returns the personal identification code found in the text. The function 'the_first_control_number_algorithm' checks if the control number of the personal identification number is correct using the first algorithm. The exercise explains that the id code contains exactly 11 numbers, and the user is asked..."
- Metadata:** null

On the right side, there is a 'Spans' panel showing a breakdown of the trace's components:

- SPAN query:** 37.06s
- SPAN retrieve:** 0.46s
- GENERATION HuggingFaceEmbedding:** 0.41s 0 → 0 (18)
- SPAN reranking:** 0.46s
- SPAN synthesize:** 36.15s
- SPAN templating:** 0.00s
- GENERATION Ollama_llm:** 36.13s

The interface also features a 'Tags' field at the top, 'Add score' and 'Add to dataset' buttons, and a 'Private' toggle switch. The bottom right corner contains a chat icon.

Joonis 11. Langfuse Trace Detail LlamaIndex query vaade.

Langfuse v2.31.0

Lucifer > Traces > 855bcd2c-6a2c-434b-a774-a17909f332c8

Trace Detail

0 → 0 (18)

Tags

SPAN query 23/04/2024, 17:46:12 Latency: 37.06s

Input: "Explain the id code exercise. Include relevant links from the context when it makes sense."

Output: "The 'id Code Exercise' is a task that requires the creation of two functions: 'find_id_code' and 'the_first_control_number_algorithm'. The function 'find_id_code' takes a given string as input and returns the personal identification code found in the text. The function 'the_first_control_number_algorithm' checks if the control number of the personal identification number is correct using the first algorithm. The exercise explains that the id code contains exactly 11 numbers, and the user is aske..."

Metadata:

```

{
  metadata: {
    metadata: {
      1988c9a0-6fab-4ef1-a988-f8ab39215dd7: {
        785302df-1e8c-4c3c-93a3-2ebc8677498f: {
        }
      }
    },
    source_nodes: [
      0: {
        node: {
          id: "1988c9a0-6fab-4ef1-a988-f8ab39215dd7"
          text: "English --- Part 1 All parts of the task go into one file in git: 'EX/ex03_idcode/idcode.py'
          Introduction The topics to be studied in the third week are: - Function - Conditional Statement - Loop Links -
          Function (PyDoc) - Conditional Statement (WSchools) - Loop (PyDoc) - Loop (Python Wikil) - National
          identification number The exercise - Write a function 'find_id_code(text: str)', which finds the personal
          identification code in the input text and returns it. The function input is a string ..."
          class_name: "TextNode"
          extra_info: {
          }
        },
        end_char_idx: 487051
        relationships: [
          1: {
            hash: "6c73a988ef838627981aeb344bb82342672ee3551c86836c7918c73d432d5ee"
            node_id: "fac7f633-e442-468b-9d6e-fc987027f95e"
            metadata: {
            }
            node_type: "4"
            class_name: "RelatedNodeInfo"
          }
        ]
      }
    ]
  }
}

```

TRACE Llamaindex_query 37.06s

- SPAN query 37.06s
 - SPAN retrieve 0.46s
 - GENERATION HuggingFaceEmbedding 0.41s 0 → 0 (18)
 - SPAN reranking 0.46s
 - SPAN synthesize 36.15s
 - SPAN templating 0.00s
 - GENERATION Ollama_llm 36.13s

Joonis 12. Langfuse Trace Detail query vaade.

Langfuse v2.31.0

Lucifer > Traces > 855bcd2c-6a2c-434b-a774-a17909f332c8

Trace Detail

0 → 0 (18)

Tags

SPAN retrieve 23/04/2024, 17:46:12 Latency: 0.46s

Input: "Explain the id code exercise. Include relevant links from the context when it makes sense."

Output:

```

{
  nodes: [
    0: {
      node: {
        id: "785302df-1e8c-4c3c-93a3-2ebc8677498f"
        text: "Mall Python ***EX03 ID code*** def find_id_code(text: str) -> str: *** Find ID-code from given text.
        Given string may include any number of numbers, characters and other symbols mixed together. The numbers of ID-
        code may be between other symbols - they must be found and concatenated. ID-code contains of exactly 11
        numbers. If there are not enough numbers, return 'Not enough numbers!'; If there are too many numbers, return
        'Too many numbers!'. If ID-code can be found, return that code. You don..."
        class_name: "TextNode"
        extra_info: {
        }
      },
      end_char_idx: 393368
      relationships: [
        1: {
          hash: "6c73a988ef838627981aeb344bb82342672ee3551c86836c7918c73d432d5ee"
          node_id: "fac7f633-e442-468b-9d6e-fc987027f95e"
          metadata: {
          }
          node_type: "4"
          class_name: "RelatedNodeInfo"
        }
      ]
    }
  ]
}

```

TRACE Llamaindex_query 37.06s

- SPAN query 37.06s
 - SPAN retrieve 0.46s
 - GENERATION HuggingFaceEmbedding 0.41s 0 → 0 (18)
 - SPAN reranking 0.46s
 - SPAN synthesize 36.15s
 - SPAN templating 0.00s
 - GENERATION Ollama_llm 36.13s

Joonis 13. Langfuse Trace Detail retrieve vaade.

The screenshot shows the Langfuse interface for a trace titled "HuggingFaceEmbedding". The main content area displays the generation details, including the input and output. The input is a JSON object with "num_chunks": 1. The output is a JSON object with "num_embeddings": 1. The generation is associated with the model "BAAI/bge-small-en-v1.5" and has a latency of 0.41s. The interface also shows a sidebar with navigation options like Dashboard, Tracing, Sessions, Generations, Scores, Models, Users, Prompts, and Playground. A right-hand sidebar shows a trace tree with steps like query, retrieve, reranking, synthesize, and templating.

Joonis 14. Langfuse Trace Detail HuggingFaceEmbedding vaade.

The screenshot shows the Langfuse interface for a trace titled "reranking". The main content area displays the generation details, including the input and output. The input is a text prompt: "Explain the id code exercise. Include relevant links from the context when it makes sense.". The output is a JSON object containing a list of nodes with detailed metadata, including IDs, hashes, and relationships. The generation is associated with the model "Ollama_lm" and has a latency of 0.46s. The interface also shows a sidebar with navigation options like Dashboard, Tracing, Sessions, Generations, Scores, Models, Users, Prompts, and Playground. A right-hand sidebar shows a trace tree with steps like query, retrieve, reranking, synthesize, and templating.

Joonis 15. Langfuse Trace Detail reranking vaade.

Langfuse v2.31.0

Lucifer > Traces > 855bcd2c-6a2c-434b-a774-a17905f332c8

Trace Detail

0 → 0 (18)

Tags

SPAN synthesize 23/04/2024, 17:46:13 Latency: 36.15s

Input

"Explain the id code exercise
Include relevant links from the context when it makes sense."

Output

"The 'id Code Exercise' is a task that requires the creation of two functions: 'find_id_code' and 'the_first_control_number_algorithm'. The function 'find_id_code' takes a given string as input and returns the personal identification code found in the text. The function 'the_first_control_number_algorithm' checks if the control number of the personal identification number is correct using the first algorithm. The exercise explains that the id code contains exactly 11 numbers, and the user is aske..."

Metadata

```

{
  "metadata": {
    "1988c9a0-6fab-4ef1-a988-f8ab39215dd7": {
      "785302df-1e8c-4c3c-93a3-2ebc8677498f": {
        "source_nodes": [
          0: {
            "node": {
              "id": "1988c9a0-6fab-4ef1-a988-f8ab39215dd7"
              "text": "---- English ---- Part 1 All parts of the task go into one file in git: 'EX/ex03_idcode/idcode.py'
              Introduction The topics to be studied in the third week are: - Function - Conditional Statement - Loop Links -
              Function (PyDoc) - Conditional Statement (W3Schools) - Loop (PyDoc) - Loop (Python Wiki) - National
              identification number The exercise - Write a function 'find_id_code(text: str)', which finds the personal
              identification code in the input text and returns it. The function input is a string ..."
              class_name: "TextNode"
              extra_info: {
            }
            end_char_idx: 487051
            relationships: [
              1: {
                "hash": "6c73a986ef858627981a0b344ab8d2342672ee30551c0e836c7918c73d432d5ee"
                node_id: "facf6533-e442-468b-9d6e-fc98702f5fae"
                metadata: {
                  "node_type": "4"
                  class_name: "RelatedNodeInfo"
                }
              }
            ]
          }
        ]
      }
    }
  }
}

```

TRACE Llamaindex_query 37.06s

SPAN query 37.06s

SPAN retrieve 0.46s

GENERATION HuggingFaceEmbedding 0.41s 0 → 0 (18)

SPAN reranking 0.46s

SPAN synthesize 36.15s

SPAN templating 0.00s

GENERATION Ollama_lm 36.13s

Project: Lucifer

AV Alex Vah...

Joonis 16. Langfuse Trace Detail synthesize vaade.

Langfuse v2.31.0

Lucifer > Traces > 855bcd2c-6a2c-434b-a774-a17905f332c8

Trace Detail

0 → 0 (18)

Tags

SPAN templating 23/04/2024, 17:46:13 Latency: 0.00s

Input

```

{
  "template": "system: You are an expert Q&A system that is trusted around the world. Always answer the query using the
  provided context information, and not prior knowledge. Some rules to follow: 1. Never directly reference the given
  context in your answer. 2. Avoid statements like 'Based on the context, ...' or 'The context information ...' or
  anything along those lines. user: Context information is below. ----- (context_str) -----
  ----- Given the context information and not prior k...\"",
  "system_prompt": null,
  "template_vars": {
    "query_str": "Explain the id code exercise
    Include relevant links from the context when it makes sense."
    "context_str": "---- English ---- Part 1 All parts of the task go into one file in git: 'EX/ex03_idcode/idcode.py'
    Introduction The topics to be studied in the third week are: - Function - Conditional Statement - Loop Links -
    Function (PyDoc) - Conditional Statement (W3Schools) - Loop (PyDoc) - Loop (Python Wiki) - National
    identification number The exercise - Write a function 'find_id_code(text: str)', which finds the personal
    identification code in the input text and returns it. The function input is a string ..."
  }
  "query_wrapper_prompt": null
}

```

Output

null

TRACE Llamaindex_query 37.06s

SPAN query 37.06s

SPAN retrieve 0.46s

GENERATION HuggingFaceEmbedding 0.41s 0 → 0 (18)

SPAN reranking 0.46s

SPAN synthesize 36.15s

SPAN templating 0.00s

GENERATION Ollama_lm 36.13s

Project: Lucifer

AV Alex Vah...

Joonis 17. Langfuse Trace Detail templating vaade.

Langfuse v2.31.0

Lucifer > Traces > 855bcd2c-6a2c-434b-a774-at7905f332c8

Trace Detail

0 → 0 (18)

Tags

GENERATION Ollama_llm
23/04/2024, 17:46:13

Latency: 36.13s 0 prompt → 0 completion (1 0) llama2:7b max_tokens: temperature: 0.75 request_timeout:

Add score + Add to dataset

Pretty JSON

```

system
"You are an expert Q&A system that is trusted around the world.
Always answer the query using the provided context information, and not prior knowledge.
Some rules to follow:
1. Never directly reference the given context in your answer.
2. Avoid statements like 'Based on the context, ...' or 'The context information ...' or anything along those lines."

user
"Context information is below. ----- English --- Part 1 All parts of the task go into one file
in git: EX/ex03_idcode/idcode.py' Introduction The topics to be studied in the third week are: - Function -
Conditional Statement - Loop Links: - Function (PyDoc) - Conditional Statement (W3Schools) - Loop (PyDoc) - Loop
(Python Wiki) - National identification number The exercise - Write a function 'find_id_code(text: str)', which
finds the personal identification code in the input ..."

assistant
"The "Id Code Exercise" is a task that requires the creation of two functions: 'find_id_code' and
'the_first_control_number_algorithm'. The function 'find_id_code' takes a given string as input and returns the
personal identification code found in the text. The function 'the_first_control_number_algorithm' checks if the
control number of the personal identification number is correct using the first algorithm. The exercise explains
that the id code contains exactly 11 numbers, and the user is aske..."

```

Metadata

```

{
  "logprobs": null
}

```

TRACE Llamaindex_query 37.06s

- SPAN query 37.06s
- SPAN retrieve 0.46s
- GENERATION HuggingFaceEmbedding 0.41s 0 → 0 (1 18)
- SPAN reranking 0.46s
- SPAN synthesize 36.15s
- SPAN templating 0.00s
- GENERATION Ollama_llm 36.13s

Settings Docs Support Feedback

Project Lucifer

AV Alex Vah...

Joonis 18. Langfuse *Trace Detail Ollama llm* vaade.