CHALMERS

GÖTEBORGS UNIVERSITET

En AI-chattbot som förenklar arbetet för studievägledare

Kandidatarbete inom Datateknik

MARKUS BÖRJESSON
GABRIELA CASTILLO AVILA
AFAF KONKABESS
NILS OLSSON
SAHAR SOLTANI
MOA-TUVA THORSTENSSON

Institution för Data- och informationsteknik
CHALMERS TEKNISKA HÖGSKOLA
GÖTEBORGS UNIVERSITET
Göteborg, Sverige 2025

Kandidatarbete 2025

En AI-chattbot som förenklar arbetet för studievägledare

MARKUS BÖRJESSON
GABRIELA CASTILLO AVILA
AFAF KONKABESS
NILS OLSSON
SAHAR SOLTANI
MOA-TUVA THORSTENSSON

GÖTEBORGS
UNIVERSITET

CHALMERS

Institution för Data- och informationsteknik
Chalmers tekniska högskola
Göteborgs universitet
Göteborg, Sverige 2025

En AI-chattbot som förenklar arbetet för studievägledare

MARKUS BÖRJESSON GABRIELA CASTILLO AVILA AFAF KONKABESS
NILS OLSSON SAHAR SOLTANI MOA-TUVA THORSTENSSON

Handledare: Niklas Broberg, Data- och informationsteknik
Medrättande lärare: Birgit Grohe, Data Science och AI, Data- och informationsteknik
Examinator: Patrik Jansson, Computing Science, Data- och informationsteknik

Kandidatarbete 2025
Institution för Data- och informationsteknik
Chalmers tekniska högskola och Göteborgs universitet
412 96 Göteborg
Telefon: 031-772 1000

Göteborg 2025

En AI-chattbot som förenklar arbetet för studievägledare
MARKUS BÖRJESSON, GABRIELA CASTILLO AVILA, AFAF KONKABESS, NILS OLSSON, SAHAR SOLTANI, MOA-TUVA THORSTENSSON
Institution för Data- och informationsteknik
Chalmers tekniska högskola och Göteborgs Universitet

Sammandrag

I takt med att studentpopulationen växer ökar behovet av tillgänglig och effektiv studievägledning. Detta kandidatarbete undersöker hur chattbottar baserade på stora språkmodeller (LLM:er), kan stödja studievägledares arbete. Genom intervjuer med studievägledare vid Göteborgs universitet framkom att svar på studenters frågor ofta redan fanns på universitetets webbplatser. Projektet inleddes med att samla in och strukturera relevant information från dessa källor genom webbsökning och automatiserad datahämtning.

För att skapa en domänspecifik chattbot baserad på denna data implementerades ett ”Retrieval-Augmented Generation”-system (RAG-system). Eftersom projektet fokuserar på att besvara vanligt förekommande frågor från studenter, undersöktes även fine-tuning av en LLM som ett alternativ. Detta gjordes genom en automatiserad process som genererade fråga-svar-par.

Försöket med fine-tuning visade inga signifikanta resultat, men har potential att ge bättre utfall vid framtida försök med mer tid och resurser. Det RAG-baserade systemet visade däremot goda resultat, men kräver vidareutveckling för att kunna implementeras i praktiken.

Nyckelord: Chattbot, RAG, fine-tuning, studievägledning, LLM.

An AI chatbot designed to facilitate the work of study guidance councilor.
MARKUS BÖRJESSON, GABRIELA CASTILLO AVILA, AFAF KONKABESS, NILS OLSSON, SAHAR SOLTANI, MOA-TUVA THORSTENSSON
Department of Computer science and engineering
Chalmers University of Technology and Gothenburgs University

Abstract

As the student population continues to grow, so does the demand for accessible and effective academic guidance. This bachelor's thesis investigates how AI-based chatbots, powered by large language models (LLMs), can be utilized to support the work of academic advisors. Through interviews with study guidance counselors at the University of Gothenburg, it was found that responses to most student inquiries are already available on the university's official websites. Accordingly, the project commenced with the collection and structuring of relevant information from these sources through automated web scraping.

To develop a domain-specific chatbot based on this data, a Retrieval-Augmented Generation (RAG) system was implemented. Given the focus on addressing frequently asked questions from students, the potential of fine-tuning a large language model was also explored. This was carried out through an automated pipeline that generated question–answer pairs for training purposes.

The attempt at using fine-tuning did not yield significant results; however, it holds potential for future experiments given more time and resources. In contrast, the RAG-based system showed promising results, although it requires further development to be practically implemented.

Key words: Chatbot, RAG, fine-tuning, study guidance, LLM.

Förord

Denna kandidatuppsats har skrivits av sex studenter vid Chalmers Tekniska Högskola under vårterminen 2025. Arbetet har handletts av Niklas Broberg, universitetslektor vid Data- och informationsteknik. Vi vill tacka vår handledare Niklas för all vägledning.

Vi vill även tacka studievägledarna som deltagit i intervjuer och de studenter som svarade på vår enkät. Andra personer som bidragit med allmänna insikter kring chattbottutveckling vill vi också tacka.

Göteborg, juni 2025

Förkortningar och begrepp

API Application Programming Interface, är ett gränssnitt som gör det möjligt för olika program att kommunicera med varandra genom att skicka och ta emot data på ett standardiserat sätt.

LLM Large Language Model, är en typ av AI som tränats på stora mängder data för att analysera och generera mänskligt språk. Den utnyttjar avancerade maskininlärningsmetoder för att förutsäga och producera text baserat på givna instruktioner eller indata.

NLP Natural Language Processing, är ett forskningsfält inom artificiell intelligens och datavetenskap som syftar till att utveckla metoder och tekniker för att möjliggöra datorers automatiserade förståelse, tolkning, bearbetning och generering av mänskligt språk i både talad och skriven form.

RAG Retrieval-Augmented Generation, är en AI-teknik som kombinerar informationssökning med textgenerering. Den hämtar relevant information från en databas eller ett dokument innan den genererar ett svar, vilket gör resultaten mer faktabaserade och kontextuella.

”First-line”-fråga är en enkel, allmän eller administrativ fråga som kan besvaras snabbt och ofta utan att kräva någon djupgående individuell bedömning. Dessa kan ofta hanteras av en chattbot eller en generell informationssida.

”Second-line”-fråga är mer komplex och kräver ofta en individuell bedömning eller tolkning. Dessa typer av frågor passar bättre för en personlig kontakt, eftersom de rör specifika situationer, mål eller problem.

Innehåll

1
Introduktion

Utbildningsnivån i Sverige har ökat markant de senaste åren, enligt SCB-statistik [1]. Allt fler söker sig till eftergymnasial utbildning, vilket bland annat har lett till en högre arbetsbelastning för studievägledare inom högre utbildning. Den senaste utvecklingen inom artificiell intelligens möjliggör emellertid framtagandet av chattbottar som i hög grad kan föra samtal av mänsklig karaktär [2]. Detta projekt undersöker hur en AI-baserad chattbot kan användas för att besvara ”first-line”-frågor som kan handla om kurser, antagningskrav, deadlines eller administrativa processer om utbildningar vid Göteborgs universitet, i syfte att minska arbetsbördan för studievägledarna.

1.1 Situationen för studievägledare

Studievägledning är ett väsentligt stöd för studenter och spelar en avgörande roll i deras akademiska och personliga utveckling [3, 4]. Studievägledare har en bred och komplex arbetsbeskrivning som innefattar att ge vägledning om utbildningsval, karriärmöjligheter och individuella studieplaner. De erbjuder stöd i beslutsfattande processer, hjälper till med frågor om behörighet och antagningskrav samt ger råd om framtida yrkesvägar. Dessutom utgör de en viktig resurs för studenter som upplever akademiska svårigheter eller behöver hjälp med att hantera psykosociala utmaningar. Det kan dock förekomma skillnader mellan olika institutioner, när det gäller ansvar, arbetsuppgifter och tillgänglighet, vilket innebär att rollen som studievägledare kan variera beroende på den specifika institutionens behov och struktur. Genom samtal kan studievägledare bidra till att minska studenters osäkerhet kring studier och öka deras motivation och engagemang.

Arbetet som studievägledare är mångsidigt, och den ständigt växande studentpopulationen ökar arbetsbelastningen [1]. En stor del av arbetet består av att besvara ”first-line”-frågor. Även om dessa frågor ofta är enkla att besvara, upptar de en betydande del av studievägledarnas tid [3].

Denna tid och energi skulle kunna användas för mer komplexa och personliga frågor, så kallade ”second-line”-frågor. Detta inkluderar individuell studieplanering, stöd till studenter med särskilda behov samt förstärkt arbete för att motverka psykosociala utmaningar bland studenter. Dessa uppgifter kräver ofta mänsklig närvaro, empati och expertis.

1.2 AI-chattbot som stöd i studievägledning

AI-baserade chattbottar har potential att avlasta studievägledare genom att hantera ”first-line”-frågor och effektivisera kommunikationen med studenter. En sådan lösning skulle inte ersätta studievägledare, utan ska snarare fungera som ett komplement som frigör tid för dem att fokusera på ”second-line”-frågor. Detta skulle inte bara öka effektiviteten utan också förbättra kvaliteten på det stöd som studenter får. Moderna AI-chattbottar bygger på stora språkmodeller (som hädanefter kommer benämnas LLM:er efter engelskans Large Language Models) och tekniker som Retrieval-Augmented Generation (RAG). Dessa tekniker gör det möjligt att ge mer precisa och relevanta svar på faktabaserade frågor, vilket är användbart vid hanteringen av återkommande ärenden inom studievägledning [5, 6].

Samtidigt finns det viktiga etiska och praktiska överväganden att ta hänsyn till innan en sådan lösning kan implementeras. Det är avgörande att en AI-lösning inte skapar en barriär för mänsklig kontakt eller ger studenter en känsla av isolering. Studenter som behöver personligt stöd eller hjälp med komplexa frågor måste fortfarande ha tillgång till mänskliga studievägledare utan långa väntetider, krångliga bokningssystem eller andra onödiga begränsningar. Dessutom behöver systemet garantera sekretess, integritet och tillförlitlighet i de svar som ges [7].

1.3 Syfte

Syftet med projektet är att undersöka hur studievägledare kan avlastas genom att en chattbot hanterar återkommande frågor, som idag upptar en betydande del av deras arbetstid. Projektet syftar även till att bygga en AI-baserad chattbot med hjälp av färdiga LLM:er för att undersöka hur en sådan chattbot kan besvara frågor om utbildningar vid Göteborgs universitet.

1.4 Undersökningsfrågor

För att uppfylla projektets syfte har följande undersökningsfrågor identifierats:

1. Vilka typer av utbildningsrelaterade frågor är möjliga och lämpliga att hantera med hjälp av ett AI-baserat chattbottssystem, givet både tekniska begränsningar och etiska riktlinjer?
2. Vilka designprinciper och tekniska komponenter krävs för att konstruera en domänanpassad chattbot som på ett tillförlitligt sätt kan besvara dessa frågor?
3. I vilken utsträckning kan nuvarande teknologier, såsom LLM:er och RAG, möjliggöra en användbar och korrekt frågehantering i denna kontext?
4. Vilka etiska överväganden behöver göras vid användning av AI i studievägledningssammanhang, särskilt i relation till studenters integritet, informationssäkerhet och tillgång till mänsklig kontakt?

1.5 Avgränsningar

För att hålla projektet fokuserat och genomförbart fastställs tydliga avgränsningar gällande chattbottens funktionalitet, teknologival och målgrupp. Dessa avgränsningar säkerställer att systemet uppfyller sitt syfte inom ramen för tillgängliga resurser och tekniska förutsättningar.

1.5.1 Avgränsningar i chattbottens funktionalitet

Chattbottens användning begränsas till att besvara återkommande, generella frågor som exempelvis ”Hur många poäng krävs för en kandidatexamen?”. Mer komplexa eller individanpassade ärenden, såsom schemaläggning, ska hanteras av behörig personal.

Begränsningen mot individrelaterade frågor är även motiverad av datatillgången: chattbotten har varken åtkomst till personuppgifter eller möjlighet att behandla dem. Därmed exkluderas frågor som kräver personlig kontext.

Vidare innebär valet av en LLM som underliggande teknik vissa inneboende begränsningar. LLM:er tenderar att generera felaktig eller påhittad information, ett fenomen känt som *hallucinationer* [8, 9, 10]. Därför krävs noggrann övervakning och granskning av genererade svar, särskilt i frågor som kräver hög informationssäkerhet eller källgranskning.

1.5.2 Teknologival och implementation

Projektet bygger på existerande LLM:er, såsom Large Language Model Meta AI (LLaMA) och OpenAI:s Generative Pre-trained Transformer (GPT), snarare än att utveckla en egen LLM från grunden [5]. Att träna en LLM kräver omfattande resurser; enligt NVIDIA behövs betydande beräkningsresurser och en budget på flera miljoner USD [11]. Med en projektbudget på 3000 kr och en arbetsstyrka om sex studenter är detta inte genomförbart. Fokus ligger därför på att anpassa befintliga LLM:er.

1.5.3 Användargrupp för chattbotten

Chattbotten är avsedd för studenter vid Göteborgs universitet, med fokus på frågor relaterade till universitetets utbildningsutbud och administrativa processer.

1.6 Tillämpning av AI i projektarbetet

Användningen av AI under projektets gång har genomförts med försiktighet, där all information som tillhandahålls av AI har noggrant kontrollerats och källor har verifierats. Vidare har AI använts för generering av Overleaf-kod(för tabeller och punktlistor), stavningskontroll, grammatikkontroll och språkliga korrigeringar för

ökad tydlighet. Här avses formateringskod i LaTeX, såsom strukturering av dokumentet och korrekt användning av kommandon. AI användes även vid implementeringen av chattbotten, exempelvis för att kunna klassificera relevanta sidor på GU.se (se avsnitt 3.1.3) och som hjälpmedel för felsökning av kod.

1.7 Språkbruk

Eftersom denna rapport är författad på svenska har svenska facktermer använts i största möjliga utsträckning. Vissa tekniska begrepp, såsom *fine-tuning* och *LLM* (Large Language Model), har emellertid bevarats i sin engelska form. Anledningen till detta är att dessa termer är väletablerade inom det aktuella forskningsfältet och ofta anses vara mer precisa och vedertagna än deras svenska motsvarigheter, såsom *finjustering* eller *språkmodell*. Vidare förekommer vissa begrepp, exempelvis *web crawling*, för vilka någon vedertagen svensk översättning inte finns. Terminologivalen har därmed gjorts i syfte att säkerställa både begreppslig tydlighet och språklig konsekvens genom hela rapporten.

2
Teoretisk bakgrund

Detta avsnitt syftar till att ge läsaren en fördjupad förståelse för den befintliga forskningen och de teknologier som använts vid utvecklingen av chattbotten.

2.1 Tidigare forskning

Utvecklingen av AI-chattbottar började på 1950-talet, när Alan Turing introducerade idén om Turingtestet, som är ett sätt att undersöka om en maskin kan kommunicera på ett mänskligt sätt [12]. Ett annat betydelsefullt steg togs år 1956 när John McCarthy organiserade en två månader lång workshop vid Dartmouth College i USA. I förslaget till denna workshop använde McCarthy för första gången termen artificiell intelligens, vilket ofta betraktas som AI:s födelseögonblick [13].

Den första betydande praktiska tillämpningen av en LLM kom år 1966 med ”ELIZA”, utvecklad av Joseph Weizenbaum [14]. Den kunde simulera en terapeut genom att upprepa och omformulera användarens inmatning. I verkligheten saknade denna LLM både språklig och semantisk förståelse.

På 1970- och 1980-talen utvecklades mer avancerade konversationssystem, såsom ”PARRY”, som simulerade en paranoid person, och ”RACTER”, som kunde generera slumpmässiga texter. Dessa system var i grunden begränsade och saknade anpassningsförmåga [15, 16].

Under 1990-talet och början av 2000-talet blev chattbottarna smartare. Ett framträdande exempel var Artificial Linguistic Internet Computer Entity (”A.L.I.C.E”), som använde Artificial Intelligence Markup Language (AIML), för att generera strukturerade svar [16].

Under 2010-talet tog utvecklingen av chattbottar stora steg framåt, tack vare utvecklingen inom artificiell intelligens och språkteknologi. Genom att använda maskininlärning och naturlig språkbehandling kunde de i högre grad efterlikna mänsklig kommunikation och som ett resultat generera ett språk som uppfattas som mänskligt och låter naturligt.

I takt med att AI och Natural Language Processing (NLP) utvecklades under 2010- och 2020-talen, förvandlades chattbottar från enkla regelbaserade system till mer avancerade AI-baserade verktyg [17]. NLP är ett delområde inom artificiell intel-

ligens som syftar till att möjliggöra för datorer att tolka och generera mänskligt språk. Området kombinerar insikter från lingvistik och maskininlärning för att minska klyftan i kommunikationen mellan människa och maskin [18]. Genom att tränas på mycket stora mängder textdata kunde LLM:er som OpenAI:s GPT-3 generera allt mer sammanhängande och flexibla dialoger [19]. Denna teknologiska utveckling har banat väg för en bred användning inom olika områden. Idag används de inom en rad olika sektorer, inklusive hälso- och sjukvård, finans och resor.

Ett exempel inom hälsoområdet är"Woebot", en chattbot inriktad på psykisk hälsa [20]. "Woebot" erbjuder känslomässigt stöd och stämningsspårning genom användning av tekniker från kognitiv beteendeterapi. Dess tillgänglighet dygnet runt kan bidra till att avlasta vården genom att erbjuda stöd vid lättare besvär och därigenom frigöra resurser för mer akuta eller komplexa behov.

Inom finanssektorn används chattbottar för kundservice och bedrägeribekämpning. "Eno", Capital Ones AI-assistent, övervakar konton för misstänkt aktivitet, svarar på fakturafrågor via SMS eller en applikation, och kan även förhandla om återbetalning för felaktiga abonnemangsavgifter [21]. Med proaktiva varningar och naturliga-språk-dialoger stärker "Eno" både användarens förtroende och bankens effektivitet.

Inom resebranschen har chattbotten "Ask Julie" hanterat över fem miljoner frågor per år sedan lanseringen [22]. Integrerad på Amtraks webbplats och app hjälper den till med biljettbokning, tidtabeller och realtidsinformation, vilket avlastar kundtjänst och förbättrar kundupplevelsen [23].

Dessa exempel visar hur moderna chattbottar revolutionerar serviceleverans. Till skillnad från tidiga system som "ELIZA" och "PARRY" kombinerar dagens bottar NLP med stora datamängder och adaptivt lärande, vilket gör dem kapabla att hantera komplexa, sektorsspecifika uppgifter och erbjuda skalbart, personligt stöd. Denna utveckling är särskilt tydlig inom utbildningssektorn, där chattbottar alltmer används för att stödja studenter såväl administrativt som pedagogiskt. Med hjälp av avancerad språkteknologi kan dessa bottar hantera en bred variation av frågor, från kursregistrering och examinationsprocesser till studiestöd och handledning, och därmed minska trycket på universitetens personal samtidigt som studenterna får omedelbar hjälp dygnet runt.

2.2 Praktiska tillämpningar av AI-assistenter inom utbildning och offentlig förvaltning

I takt med den teknologiska utvecklingen inom artificiell intelligens har områdesspecifika AI-baserade chattbottar blivit allt vanligare inom flera sektorer, inklusive utbildningsväsendet. Flera lärosäten har på olika sätt integrerat dessa teknologier för att förbättra informationsspridning och administrativa processer.

Ett tidigt exempel är Georgia State Universitys chattbot "Pounce", som introdu-

cerades år 2016 i syfte att motverka det så kallade *summer melt*-fenomenet, där antagna studenter inte slutför sin registrering [24]. Genom att besvara vanliga frågor via SMS, såsom frågor om studiestöd, boende och kursval, lyckades universitetet avsevärt öka andelen studenter som fullföljde sin inskrivning.

Vid vissa akademiska institutioner har AI-baserade chattbottar utvecklats inom ramen för forskningsprojekt innan de implementerats i praktiken. Ett ofta citerat exempel är ”Jill Watson”, en virtuell undervisningsassistent utvecklad vid Georgia Institute of Technology [25]. Projektet initierades som en forskningsstudie och testades därefter i verkliga onlinekurser, där den virtuella assistenten besvarade studentfrågor i kursforum baserat på noggrant granskad och tillförlitlig kursinformation.

Ytterligare ett exempel återfinns vid Illinois Institute of Technology (IIT), där en FAQ-baserad chattbot har utvecklats för att bistå fakultetens personal i frågor rörande forskningsansökningar och projektadministration [26]. Systemet är integrerat med universitetets forskningsstödsenhet (Office of Research Support Programs, ORSP) och tillhandahåller information om anslagsansökningar, post-award-processer samt andra relaterade ärenden. Målet har varit att erbjuda snabba och relevanta svar, även i en föränderlig regulatorisk kontext.

Även inom Sverige har vissa lärosäten implementerat chattbottar som ett verktyg för att effektivisera kommunikationen mellan studenter och administration. Dessa initiativ är i regel inte sprungna ur forskningsverksamhet utan har ofta karaktären av IT-projekt. Ett exempel är Linnéuniversitetets digitala assistent ”Calle”, som fungerar som en virtuell guide och besvarar övergripande frågor relaterade till studier, inklusive antagningsinformation och programutbud [27]. Användarna uppmanas att ställa en fråga i taget för att underlätta systemets tolkning, och tjänsten är kontinuerligt tillgänglig via universitetets webbplats. Trots denna tillgänglighet har ”Calle” identifierade begränsningar. Svaren är ofta generella och består i hög grad av hänvisningar till omfattande listor av webblänkar, snarare än specifika och kontextanpassade svar. Vid mer detaljerade frågor genererar systemet vanligtvis flera länkförslag snarare än ett direkt svar, vilket kan försvåra informationssökningen. Om relevant information saknas, efterfrågar systemet en omformulering, föreslår irrelevanta resurser eller hänvisar till IT-support, bibliotek eller övriga servicefunktioner.

Sundsvalls kommun har implementerat en AI-baserad assistent på sin plattform för vuxenutbildning [28]. Denna assistent besvarar frågor relaterade till kursanmälan, ansökningstider och kontaktuppgifter, samt erbjuder direktlänkar till kurskatalogen för vidare information. Syftet är att avlasta personalen och förbättra tillgängligheten för medborgarna. AI-assistenten i Sundsvall tillhandahåller strukturerade svar, ofta i form av punktlistor med relevanta länkar. Den klarar av att hantera både specifika och generella frågeställningar. Vid otillräcklig information informeras användaren om detta, samtidigt som det klargörs vilka typer av frågor systemet kan hantera. I dessa fall länkas användaren ofta vidare till kompletterande resurser såsom studievägledare, IT-support eller andra relevanta kontaktpunkter inom vuxenutbildningen. En begränsning med systemet är att det inte tillhandahåller detaljerad information

om enskilda kurser eller deras specifika förkunskapskrav.

Luleå kommun har lanserat chattbotten ”SYV-Kim” som en digital medarbetare inom vuxenutbildningen [29]. ”SYV-Kim” vägleder användare i frågor om kursutbud, ansökningsprocesser och kontaktmöjligheter och är integrerad i kommunens utbildningsportal. Systemet fungerar som ett kompletterande stöd till traditionell studievägledning och bidrar till en mer effektiv kommunikationsprocess mellan medborgare och utbildningsorganisation. ”SYV-Kim” uppvisar för närvarande flera begränsningar. Svaren är ofta generella och består i stor utsträckning av hänvisningar till externa informationssidor, snarare än specifika och direkt användbara svar i chatten. I många fall krävs det att användaren följer rekommenderade länkar för att få tillgång till fullständig information. Vid mer komplexa eller otydliga frågor uppmanas användaren att omformulera sin fråga och systemet kan även i de fall hänvisa till kommunens kontaktuppgifter.

2.3 Large Language Models (LLM:er)

Stora språkmodeller (LLM:er) är avancerade AI-system som har förmågan att bearbeta och generera text med hög grad av sammanhang, semantisk relevans och språklig koherens. Dessa LLM:er har uppvisat betydande potential inom ett brett spektrum av uppgifter relaterade till området NLP.

En central komponent vid träning av LLM:er är tokenisering, en process där text delas upp i mindre beståndsdelar kallade tokens [30]. Dessa tokens kan utgöras av enskilda tecken, symboler eller ord. Genom användning av uppmärksamhetsmekanismer, framför allt self-attention, kan en LLM prioritera tokens utifrån deras relativa betydelse inom den givna kontexten.

De arkitekturer som ligger till grund för LLM:er är komplexa, och kräver flera utvecklingssteg samt avancerad språkförståelse. En avgörande metod för att möjliggöra denna förståelse är användningen av så kallade *embeddings*, där ord representeras som vektorer i ett flerdimensionellt rum [31]. I denna representation positioneras ord som ofta förekommer i liknande sammanhang nära varandra, vilket gör det möjligt för modellen att identifiera semantiska likheter. Exempelvis kommer orden ”hund” och ”valp” att placeras nära varandra i detta rum, då de bär på liknande betydelser. Denna struktur möjliggör för LLM:er att upptäcka lingvistiska mönster samt förutsäga efterföljande sekvenser i en text.

För att ytterligare förstärka kontextförståelsen och relationerna mellan ord, tillämpas positionskodning. Denna teknik möjliggör för LLM:er att särskilja mellan meningar som ”Du gillar glass” och ”Gillar du glass”, trots identiska ordkomponenter. Här spelar ordens relativa position en avgörande roll för meningsinterpretationen.

En annan väsentlig komponent i LLM:ers arkitektur är viktningsmekanismen, där specifika ord tilldelas varierande grad av uppmärksamhet beroende på deras betydelse i kontexten. Detta sker huvudsakligen genom self-attention. Mekanismen är

av särskild betydelse vid tolkning av meningar där ett enskilt ord kan förändra innebörden drastiskt, såsom skillnaden mellan ”Du gillar glass” och ”Du gillar **inte** glass”. För chattbottar är förmågan att uppfatta och hantera sådana språkliga nyanser avgörande för att kunna generera meningsfulla och kontextuellt korrekta svar, och därigenom undvika det statiska och mekaniska intryck som kännetecknade tidigare generationer av chattbottar.

Trots de framsteg som gjorts inom LLM-utveckling, återstår ett flertal begränsningar och riskmoment. En av de mest påtagliga utmaningarna är beroendet av omfattande datamängder [31]. Dessa datamängder utgör grunden för LLM:ers funktionalitet, men kräver kontinuerlig uppdatering och underhåll i takt med att information förändras över tid. Detta är inte bara resurskrävande utan även kostsamt. Därtill råder ofta en felaktig uppfattning om att LLM:er ”förstår” det innehåll de bearbetar. I realiteten bygger deras funktion på statistisk mönsterigenkänning snarare än genuin förståelse, vilket gör att de har begränsad kapacitet att tolka exempelvis sarkasm, ironi eller emotionella nyanser – aspekter som är centrala i mänsklig kommunikation.

En särskilt problematisk egenskap hos LLM-baserade chattbottar är förekomsten av så kallade hallucinationer, det vill säga generering av felaktig eller påhittad information [32]. Detta fenomen underminerar modellernas tillförlitlighet, vilket kan få allvarliga konsekvenser inom tillämpningsområden där korrekt information är kritisk – såsom inom medicinsk rådgivning. Hallucinationer utgör därför en central utmaning vid implementering av LLM-teknik. För att minska risken för sådana fel är det avgörande att införa kontrollmekanismer och säkerställa att modellen stöds av tillförlitliga, aktuella och verifierbara datakällor.

2.3.1 Diverse prompt tekniker

En *prompt* utgör en instruktion eller fråga som tillhandahålls av en LLM i syfte att generera text eller annan typ av respons [33]. Utan en prompt saknar modellen möjlighet att operera, då den förlitar sig på specificerade instruktioner för att förstå användarens avsikt. Vid kommunikation med en chattbot inkluderas även en fördefinierad systemprompt, vilket medför att det totala antalet tokens som skickas till LLM:en ökar. Promptens längd påverkar därmed användningskostnaden, eftersom fler tokens resulterar i högre beräkningsmässiga utgifter. Det är därför av vikt att utforma promptar som är både informativa och koncisa.

Inom området *prompt engineering* finns flera strategier som syftar till att förbättra träffsäkerheten i de svar som genereras av en LLM [33]. En vanligt förekommande strategi i chattbaserade tillämpningar är *conversational prompting*, vilken möjliggör en fortlöpande dialog där tidigare kontext bevaras och kan refereras till.

Conversational prompting innefattar ett antal tekniker för att ytterligare höja svarens precision. En sådan teknik är *roll prompting*, där modellen tilldelas en specifik roll tillsammans med en beskrivning av dess kompetenser och uppdrag [34]. Empiriska studier har visat att denna metod kan öka svarens noggrannhet med upp till 25%.

En annan effektiv teknik är *step-by-step prompting*, där en LLM instrueras att följa en detaljerad och sekventiell process [35]. En sådan struktur kan exempelvis implementeras som en punktlista som tydligt anger uppgifternas ordning och innehåll.

Genom att kombinera flera av ovanstående tekniker kan LLM:ens effektivitet förbättras [33]. Exempelvis kan *roll prompting* kombineras med *step-by-step prompting* genom att tilldela en chattbot rollen som kundtjänstmedarbetare samtidigt som den förses med en punktlista över instruktioner att följa. Denna kombinerade strategi har visat sig öka svarens precision, minska förekomsten av hallucinationer och förbättra LLM:ens funktionalitet i praktiska tillämpningar. Exempelvis kan en sådan prompt utformas på följande sätt:

> Du jobbar med kundservice på ett modeföretag, följ dessa steg tydligt för att säkerställa att du ger den bästa möjliga tjänsten;
> 1. Hälsa på kunden och se till att svara på alla frågor de kan ha.
> 2. Föreslå olika produkter baserad på kundens behov och tillgängliga produkter i butiken.
> 3. Berätta vart de kan hitta mer information om vidare assistans behövs.

2.4 Retrieval-augmented generation (RAG)

RAG har utvecklats som en lösning på de utmaningar som LLM:er står inför, särskilt gällande informationens relevans och tillgång till uppdaterad data [6]. Tekniken introducerades under 2020 och har sedan dess inneburit betydande framsteg för befintliga LLM:er. En av de huvudsakliga fördelarna med RAG är dess förmåga att minska risken för hallucinationer, samtidigt som den möjliggör att LLM:er kan hålla sig aktuella genom att hämta information från externa källor.

Traditionella LLM:er är begränsade till den data de ursprungligen tränats på, vilket innebär att integration av ny information kräver omfattande träning. RAG adresserar denna begränsning genom att införa en mekanism där LLM:er kan söka och hämta relevant information i realtid. Detta minskar behovet av att kontinuerligt träna om dem och ökar deras flexibilitet när det gäller att hantera dynamiskt förändrad information.

I ett system som implementerar RAG-teknik skickas användarens förfrågan parallellt till både en retriever och en generator (LLM). Retriever-komponenten söker efter relevant information i en extern databas och tillhandahåller detta material till en LLM, som därefter använder det som grund för att generera ett svar. Genom att kombinera retrieval och generator minskas risken för hallucinationer som annars kan uppstå när en LLM enbart baserar sina svar på en statisk och begränsad datamängd [6].

De externa källor som används av retrieval-komponenten kan exempelvis bestå av

webbsidor, PDF-dokument eller annan dokumentation. Detta gör det möjligt att anpassa en chattbotts databas till aktuell och specifik information som ursprungligen inte fanns tillgänglig för LLM:en vid dess träningstillfälle [36].

För att ett RAG-system effektivt ska kunna identifiera relevant information är embeddings en central komponent. Kvaliteten på dessa embeddings är avgörande för retrieval-komponentens prestanda, eftersom bristfälliga eller felaktiga vektorrepresentationer kan leda till irrelevanta sökresultat. För att mäta likheten mellan olika vektorer används flera matematiska metoder, där cosinuslikhet (cosine similarity) är ett vanligt förekommande exempel [36]. Retriever-komponentens huvudsakliga uppgift är att, med hjälp av dessa metoder, identifiera de mest relevanta resultaten baserat på användarens förfrågan.

2.4.1 Potentiella datasäkerhetsrisker med RAG-system

Arbetet med RAG-system innebär både betydande fördelar och potentiella nackdelar, inklusive vissa säkerhetsrisker. En central utmaning är hanteringen av känslig information, där risken för dataläckage från de externa datakällor som används är påtaglig. En studie som undersökte hur RAG-system reagerar på olika typer av attacker visade att systemen var sårbara för dataintrång [37]. I studien genomfördes tester utifrån ett black-box-perspektiv, vilket innebar att angriparna ställde frågor till en LLM utan tillgång till dess interna struktur eller funktionalitet. Angreppen fokuserade på att styra retrievern till att hämta känslig information, och därefter få en LLM att vidarebefordra denna information till angriparen genom användning av specifika och riktade promptar. Resultaten visade att RAG-baserade system är mottagliga för denna typ av attacker. För att minska risken för informationsläckage har flera åtgärder föreslagits, bland annat re-ranking. Re-ranking innebär att ytterligare LLM:er används för att rangordna den hämtade informationen utifrån dess relevans. Studien visade dock att re-ranking inte hade någon påvisbar effekt på att motverka informationsläckage [37].

Dessa typer av attacker är emellertid inte exklusiva för RAG-baserade system, utan kan även riktas mot LLM:er utan RAG-integration. Vid attacker mot traditionella LLM:er kan en liknande black-box-metodik tillämpas, där angriparens mål kan vara att få en LLM att generera felaktig eller oönskad information, alternativt att extrahera delar av deras träningsdata [38].

Dessa resultat understryker vikten av att införa robusta säkerhetsprotokoll för att hantera risker relaterade till både RAG-system och LLM:er. Trots de identifierade säkerhetsutmaningarna erbjuder RAG-teknologin betydande fördelar. Genom de mekanismer som RAG tillhandahåller kan chattbottar utvecklas för specifika tillämpningsområden utan behov av att träna om en LLM från grunden. Ett exempel på en sådan tillämpning är en chattbot för studievägledning, byggd på en vektoriserad databas innehållande relevant information om program, kurser, antagningskrav och andra studierelaterade uppgifter vid ett specifikt universitet eller lärosäte.

2.4.2 Metoder för att utvärdera RAG-system

Det finns flera metoder för att utvärdera hur effektivt ett RAG-system är, vilka ofta innefattar en kombination av automatiska mätvärden, mänskliga bedömningar och kontextspecifik analys. För informationsbaserade chattbottar är det effektivt att använda så kallade ”ground truths”, det vill säga förväntade eller korrekta svar, för att jämföra bottens svar mot ett facit [39]. Detta gör det möjligt att automatiskt mäta precision, relevans och korrekthet.

2.5 Fine-Tuning

Fine-tuning utgör en central och avgörande process inom maskininlärning, särskilt i sammanhang som rör LLM:er och chattbottar. Processen innebär att en tidigare LLM, som har tränats på omfattande och varierande datamängder från exempelvis internet, genomgår en fine-tuning med hjälp av mindre och mer specialiserade dataset. Syftet med denna ytterligare träning är att förbättra LLM:er prestanda inom en specifik uppgift eller ett avgränsat domänområde, såsom medicin, juridik eller teknisk support. På detta sätt fungerar fine-tuning som en länk mellan generell språkförståelse och specialiserad, uppgiftsanpassad förmåga [40, 41].

Moderna LLM:er som BERT och GPT bygger på transformer-arkitekturer och tränas med självövervakad inlärning [40, 42]. Transformers revolutionerar användningen av NLP och är en nätverksarkitektur för sekvenstraduktion, som förlitar sig på uppmärksamhetsmekanismen [43]. BERT använder masked language modeling (MLM), där en LLM förutsäger saknade ord i en mening, och next sentence prediction (NSP), som testar om en mening logiskt följer efter en annan, medan GPT förutspår nästa ord i en sekvens (causal language modeling) [40, 42, 44]. Trots att dessa LLM:er lär sig generella språkliga samband, saknar de ofta domänspecifik kunskap. Fine-tuning med specialiserad data kan därför öka deras träffsäkerhet och kontextförståelse [45].

För fine-tuning av LLM:er finns det två huvudsakliga tillvägagångssätt, som båda syftar till att anpassa deras beteende till specifika uppgifter eller användarkrav. Det första är Supervised Fine-Tuning (SFT) där LLM:er tränas på manuellt annoterade dataset med korrekta svar [46]. Genom att justera sina interna parametrar för att minska avvikelsen mellan de egna förutsägelserna och de angivna riktiga svaren, förbättrar en LLM sin precision och förmåga att lösa uppgiften på ett målinriktat sätt. Det andra tillvägagångssättet är Reinforcement Learning with Human Feedback (RLHF) [47]. Här används mänskliga bedömningar för att rangordna LLM:er olika svarsalternativ, vilket gör det möjligt att styra träningen mot svar som människor upplever som mer relevanta, sammanhängande och hjälpsamma. Denna metod har visat sig särskilt effektiv i utvecklingen av moderna LLM:er som ChatGPT.

Fine-tuning av LLM:er kan ske på olika sätt beroende på hur stor del av deras inre parametrar som uppdateras under träningen. Ett av de mest kraftfulla men också resurskrävande tillvägagångssätten är Full Fine-Tuning, där samtliga parametrar i LLM:er tränas om. Detta innebär att hela nätverket anpassas till den nya

uppgiften, vilket ofta leder till mycket goda resultat när tillräckligt med data och beräkningsresurser finns tillgängliga. Ett konkret exempel är fine-tuning av BERT på det målorienterade dialogdatasetet MultiWOZ [48]. Samtidigt innebär denna metod en risk för överanpassning, särskilt när mängden träningsdata är begränsad, och kräver dessutom betydande datorkraft [49].

För att hantera dessa utmaningar har forskare utvecklat mer resurssnåla metoder, samlade under begreppet Parameter-Efficient Fine-Tuning (PEFT). Dessa metoder håller majoriteten av LLM:er parametrar frysta, och endast mindre, specifika delar tränas om. Ett exempel är Low-Rank Adaptation (LoRA), där små adaptermoduler läggs till och tränas istället för att justera hela LLM:er [50]. Ytterligare ett alternativ är prefix tuning, som introducerar träningsbara tokens i början av indata för att påverka LLM:er beteende utan att ändra dess ursprungliga vikter [51].Dessa är bara några av många metoder inom PEFT som möjliggör effektiv anpassning av stora språkmodeller.

Trots fördelarna finns det flera utmaningar förknippade med fine-tuning. En av de största är bristen på högkvalitativa, domänspecifika dataset. Att samla in och annotera sådan data är tidskrävande och kostsamt. Dessutom måste man vara noga med att undvika bias och skydda användarnas integritet [52]. En annan utmaning är det som kallas catastrophic forgetting, vilket innebär att en LLM under fine-tuning glömmer bort kunskap som den tidigare hade lärt sig [53]. Överanpassning är också ett vanligt problem, särskilt när fine-tuning görs på små dataset. Det innebär att en LLM blir för bra på just träningsdatan och presterar dåligt på nya, okända exempel. För att undvika detta används metoder som regularisering och korsvalidering.

2.5.1 Metoder för att utvärdera fine-tuning-system

Att utvärdera kvaliteten på en chattbot baserad på fine-tuning är en komplex utmaning, eftersom traditionella mått inte alltid speglar hur naturlig eller meningsfull en konversation faktiskt upplevs av en mänsklig användare. Automatiska mått som BLEU och ROUGE används för att jämföra genererade texter med referenstexter genom n-gram-överlappning [54, 55]. Ett n-gram är ett sekvens av n ord. Till exempel är ”Jag gillar glass” ett 3-gram. Inom NLP används n-gram-modeller för att uppskatta sannolikheten för nästa ord i en sekvens, givet det föregående ordet [56]. BLEU mäter hur mycket av den genererade texten som matchar referenstexten, medan ROUGE fokuserar på hur mycket av referenstexten som fångas upp. De fungerar väl för uppgifter som maskinöversättning och sammanfattning, men är ofta otillräckliga för öppen dialog, där flera olika svar kan vara lika giltiga.

2.6 Pythonbibliotek som använts i projektet

Denna sektion presenterar de mest centrala biblioteken som används inom ramen för detta projekt och beskriver kortfattat deras huvudsakliga funktion samt betydelse i relation till projektets syfte och tekniska krav.

2.6.1 LangChain

LangChain är ett ramverk som syftar till att förenkla utvecklingen av olika applikationer baserade på LLM. Det tillhandahåller olika verktyg som utökar potentialen hos dem. Dessa verktyg gör det möjligt att koppla LLM:er med externa datakällor och funktioner, och möjliggör en mer flexibel och kraftfull utvecklingsmiljö [57]. LangChain erbjuder färdiga moduler samt ett brett utbud av instruktioner och handledningar för att kunna implementera både RAG och chattbottar [58, 59]. LangChain har även inbyggt stöd för att kunna hantera embeddings och vektorbaserad informationssökning.

2.6.2 BeautifulSoup

BeautifulSoup är ett Python-bibliotek som används för att extrahera och manipulera data från HTML- och XML-dokument [60]. Det erbjuder en lättanvänd struktur för att parsa dokument och tillåter programmerare att navigera i, söka efter och modifiera innehållet genom ett trädliknande applikationsprogrammeringsgränssnitt (API), vilket motsvarar engelskans ”application programming interface”. BeautifulSoup hanterar automatiskt felaktigt formaterad kod, vilket gör det särskilt användbart vid web scraping från icke-homogena källor. BeautifulSoup är därmed ett centralt verktyg inom dataextraktion och informationsåtervinning på webben.

2.6.3 RapidFuzz

RapidFuzz är ett optimerat Python-bibliotek för fuzzy matching av text, utvecklat för att kombinera hög prestanda med flexibilitet i textanalys. Biblioteket implementerar avancerade algoritmer för beräkning av likheter och stöder ordbaserad matchning där ordens inbördes ordning inte påverkar resultatet. RapidFuzz erbjuder även funktioner för regex-baserad matchning, vilket möjliggör flexibel identifiering av mönster i text. Det används främst inom datarensning, informationssökning och naturlig språkbehandling, där snabb, exakt och skalbar textjämförelse är av central betydelse.

2.6.4 Trafilatura

Trafilatura är ett Python-bibliotek utvecklat för att extrahera och strukturera innehåll från webbsidor, särskilt nyhetssidor och artiklar [61]. Det är designat för att effektivt isolera textbaserat innehåll och eliminera onödiga element. Genom att tillämpa avancerade tekniker för webbskrapning och innehållsfiltrering kan Trafilatura producera rena och välstrukturerade dokument som är lämpliga för vidare bearbetning, som textanalys eller informationssökning.

2.7 Vektoriserade databaser

ChromaDB är en lokal, öppen källkodslösning som är enkel att installera och integrera [62]. Den används ofta i prototypstadier tack vare dess låga tröskel och snabba

uppsättning. Dokument och vektorer lagras direkt på filsystemet.

AstraDB är en molnbaserad, skalbar databaslösning utvecklad av DataStax [63]. Den erbjuder ett mer robust och distribuerat alternativ, vilket lämpar sig för produktion eller större datamängder. AstraDB stödjer vektorbaserad sökning och bygger på Apache Cassandra, vilket är en distribuerad NoSQL-databas optimerad för hög tillgänglighet, skalbarhet och hantering av stora datavolymer [64].

Tabell 2.1: Jämförelse mellan ChromaDB och AstraDB

Egenskap	**ChromaDB**	**AstraDB**
Installation	Lokal, inga externa krav	Kräver molnkonto hos DataStax
Skalbarhet	Begränsad till lokal miljö	Hög, distribuerad lagring
Lagring	På användarens filsystem	I molnet
Prestanda	Bra för mindre dataset	Bättre för större datamängder
Användningsfall	Prototyper, testmiljöer	Produktionsmiljöer, skalbarhet

3
Metod

Detta arbete har huvudsakligen fokuserat på praktisk utveckling av en domänanpassad chattbot för Göteborgs universitet. Initialt genomfördes en undersökning av hur en chattbot skulle kunna utvecklas för att stödja studievägledare. Därefter följde en serie implementationer för att empiriskt utvärdera olika tekniska angreppssätt. Huvudfokus låg på utveckling av ett RAG-baserat system, men även en fine-tunad LLM undersöktes som alternativ.

I detta kapitel redogörs för de centrala metodologiska komponenterna i arbetet samt hur dessa valdes, tillämpades och vid behov anpassades. Vad gäller de tekniska metoderna har dessa utvecklats genom en iterativ process med kontinuerliga förbättringar.

3.1 Informationsinsamling

Informationsinsamlingen i detta arbete bestod av två kompletterande spår. Det första spåret utgjordes av intervjuer med studievägledare samt enkäter riktade till studenter. Syftet med dessa insatser var att erhålla ett underlag för att undersöka på vilket sätt en potentiell chattbot skulle kunna avlasta studievägledarnas arbete, samt att skapa förståelse för studenters inställning till att använda ett sådant verktyg.

Det andra spåret omfattade en systematisk insamling av information från webbaserade informationskällor (web crawling och web scraping). Denna insamling syftade till att skapa ett tillförlitligt faktaunderlag som kunde användas för att säkerställa att den utvecklade chattbotten genererar korrekta och relevanta svar baserade på existerande information.

3.1.1 Intervjuer med studievägledare

Som underlag för flera metodologiska val i arbetet genomfördes intervjuer med studievägledare. Totalt intervjuades två studievägledare vid två tillfällen vardera. Urvalet av intervjupersoner baserades inte på representativitet i förhållande till Göteborgs universitet som helhet, utan valdes utifrån tillgänglighet och tidigare kontakt med projektets handledare, Niklas Broberg. Båda intervjupersonerna var verksamma vid institutionen för data- och informationsteknik.

De inledande intervjuerna var strukturerade och genomfördes med stöd av ett in-

tervjuprotokoll som deltagarna fick ta del av i förväg. Intervjuprotokollet återfinns i bilaga A.2. Frågorna fokuserade främst på vilka typer av studentfrågor som bedömdes vara lämpliga respektive olämpliga att besvaras av en AI-baserad chattbot. En annan central aspekt var vilken typ av fakta studievägledarna vanligtvis förlitar sig på i sitt arbete, eftersom denna information utgjorde ett nödvändigt underlag för att kunna påbörja utvecklingen av en datamängd.

De uppföljande intervjuerna hade en explorativ och semistrukturerad karaktär. Under dessa intervjuer fick studievägledarna interagera med en prototyp av chattbotten och därefter ge kvalitativ återkoppling på dess svar. Särskild vikt lades vid att identifiera vilka svar som uppfattades som relevanta, informativa och korrekta, samt vilka aspekter som upplevdes som bristfälliga eller missvisande.

3.1.2 Enkät riktad till studenter

För att kartlägga studenters erfarenheter, metoder för informationssökning och attityder gentemot studievägledning samt AI-baserade chattbottar genomfördes en enkätundersökning. Enkäten riktades till studenter vid Chalmers tekniska högskola och Göteborgs universitet och besvarades av totalt 81 deltagare. Frågorna i enkäten berörde flera områden, såsom hur studenter söker studierelaterad information, deras tidigare kontakt med studievägledare, vilka typer av frågor de har haft, samt deras inställning till att använda AI-teknik i form av chattbottar för vägledning och information.

Enkäten var tillgänglig på både svenska och engelska för att nå så många studenter som möjligt oavsett språkbakgrund. Den distribuerades digitalt och deltagandet var frivilligt och anonymt. Resultaten analyserades deskriptivt för att identifiera mönster, vanliga behov och eventuella utmaningar kopplade till studierelaterad information och vägledning (se figur A.1–A.4).

3.1.3 Web Crawling

Intervjuer med studievägledare visade att frågor av saklig karaktär i regel besvaras med information hämtad från universitetets webbplatser. Den specialiserade kunskap som chattbottarna i detta projekt tillhandahåller har därför uteslutande extraherats från webbsidor och PDF-dokument tillgängliga under domänerna https://www.gu.se/ och https://studentportal.gu.se/.

Syftet med detta moment var att identifiera potentiellt relevanta webbresurser inom ovan nämnda domäner, i syfte att möjliggöra informationsutvinning relaterad till studieadministrativa frågor. En omfattande ”web crawling”-process genomfördes, vilket resulterade i 75 053 identifierade URL:er (den 25 februari 2025). Vidare identifierades ett API, från vilket 11 767 kurs- och programplaner extraherades på svenska, varav 3 532 även fanns tillgängliga på engelska.

Eftersom en majoritet av URL:erna bedömdes vara irrelevanta i sammanhanget, ge-

nomfördes en flerstegsfiltrering. Irrelevanta sidor ökar både beräkningskostnaderna och risken för inkorrekt informationsmatchning i senare steg.

Inledningsvis eliminerades dubbletter av webbsidor där den enda skillnaden låg i användningen av http respektive https, genom att behandla dessa som samma adress. Därefter uteslöts URL:er som innehöll något av följande nyckelord:

nyheter, aktuellt, forskning, evenemang, hitta-person, konferens, litteraturlista, nationella-prov, news, event, research, conference, find-staff, reading-list, node, core-facilities, docx

Denna regelbaserade filtrering minskade antalet sidor till 14 455. Därefter användes en semantisk metod för att minska mängden ytterligare. Från huvudinnehållet – efter att menyer och liknande element tagits bort – extraherades tre utdrag om 400 tecken vardera från början, mitten och slutet av varje sida eller PDF-dokument. Om texten understeg 1 200 tecken användes hela innehållet. Dessa utdrag analyserades med hjälp av en LLM (gpt-4o-mini-2025-02-15 från OpenAI). Modellen instruerades att bedöma om innehållet var relevant för en studievägledare som besvarar studenters frågor.

Exempel på relevanta teman är information om utbildningar och kurser, antagningskrav och studiestöd, samt tjänster och kontaktvägar riktade till studenter. Typiska exempel inkluderar sidor om programutbud, tentamensregler eller stipendiemöjligheter. Icke-relevant innehåll omfattade exempelvis forskningsmaterial utan studentperspektiv, interna nyheter för personal, upphandlingar eller pressmeddelanden utan koppling till studenter. Även subjektiva texter, som intervjuer med studenter, bedömdes som mindre relevanta. Modellen ombads att klassificera varje sida med ett av tre svar: *ja*, *nej* eller *kanske*. Den fullständiga prompten återfinns i B.1.

Endast sidor som klassificerades som *ja* eller *kanske* behölls, vilket minskade mängden URL:er till 5 077. Prompten var formulerad på engelska, oavsett vilket språk den aktuella sidan hade.

3.1.4 Web scraping

Många av de analyserade webbsidorna innehöll återkommande, irrelevant metadata såsom menyer, sidfötter och navigeringslänkar. Biblioteket Trafilatura (se avsnitt 2.6.4) testades initialt för extraktion av huvudtext, men visuell granskning visade att standardinställningarna var alltför restriktiva och ofta uteslöt relevant innehåll.

Ett anpassat skript utvecklades därför med hjälp av biblioteket BeautifulSoup (se avsnitt 2.6.2). Skriptet extraherade innehåll från `<article>`-taggar och tillhörande text, varefter återkommande generisk information i början och slutet av sidorna filtrerades bort.

3.2 RAG-baserad chattbot

Utvecklingen av det RAG-baserade systemet bestod av två huvudkomponenter: dels en rent vektorbaserad informationsåtervinning, dels ett komplementärt system med regelbaserad ordmatchning specifikt framtaget för effektivare återgivning av kurs- och programinformation.

Materialet som extraherades från webbsidor och PDF-dokument segmenterades i mindre informationsenheter, benämnda *chunks*. Varje chunk transformerades till en numerisk vektorrepresentation (embedding) med bibehållen källa (URL), och dessa lagrades i en vektordatabas (vector store) för senare användning vid semantisk informationssökning.

Parallellt användes en strategi för explicit hantering av kurs- och utbildningsplaner. Genom ordmatchning av kurs- och programnamn samt koder identifierades relevanta dokument. Vid positiv matchning extraherades och tillhandahölls dessa dokument till LLM:er analogt med chunks från vector store. I de fall där information inte kunde lokaliseras i en specifik plan föll systemet tillbaka på vektorsökning.

3.2.1 Chunking och embedding

En betydande del av det innehåll som publiceras på universitetets webbsidor och i PDF-dokument är strukturerat enligt rubrikhierarkier. Innehållets semantiska innebörd är därmed beroende av denna struktur. Ett exempel är kursplaner där rubriken som anger kursens identitet är avgörande för förståelsen av efterföljande sektioner. Till exempel blir följande, vanligt förekommande, formulering meningslös utan information om vilken kurs som avses:

> **Förkunskapskrav**
> *För tillträde till kursen krävs grundläggande behörighet.*

För att bevara denna kontext implementerades en hierarkisk chunkningsstrategi, där varje textstycke annoterades med sin fullständiga rubrikstig. För webbsidor utnyttjades `<h1>`–`<h6>`-taggar. I PDF-dokument antogs rubriker vara satta i fetstil och att en mer övergripande rubrik hade ett större typsnitt, vilket visade sig stämma väl med universitetets dokumentstandard.

Under chunkningen hanterades dokumenten sekventiellt uppifrån. Rubrikerna placerades i en stack; när brödtext påträffades slogs stackens innehåll samman med textstycket för att bilda en chunk. Nya överordnade rubriker triggade att stacken spolades tillbaka i enlighet med den hierarkiska ordningen.

Alla chunks transformerades med LLM:en `text-embedding-3-small` från OpenAI. Dessa embeddings tillsammans med motsvarande URL:er lagrades i en ChromaDB-instans för utveckling. I ett senare skede migrerades datan till en AstraDB-instans för att utvärdera systemets skalbarhet.

3.2.2 Routing

En viktig del av systemet var frågeklassificering, så kallad *routing*. För att möjliggöra olika bearbetningsvägar beroende på frågetyp användes en LLM som instruerades att klassificera frågor med ett begränsat antal fördefinierade svarsalternativ. Resultatet tolkades för att styra frågan till rätt hanteringsprocess. Implementationen skedde med hjälp av biblioteket LangChain (se avsnitt 2.6.1).

3.2.3 Matchning av kursnamn och kurskoder

För att matcha studenters frågor med korrekt kurs- eller programinformation krävdes tillgång till aktuella listor över kurs- och programnamn samt tillhörande koder. Två metoder utvärderades: en manuell regelbaserad ”fuzzy”-matchning och en semantisk sökmetod i en vektorbaserad databas.

Den manuella ”fuzzy”-matchningen implementerades med hjälp av biblioteket RapidFuzz (se avsnitt 2.6.3). Varje kursnamn och -kod, samt programnamn och programkod testades mot studentens fråga med hjälp av flera olika likhetsmått tillgängliga i RapidFuzz-biblioteket. De tolv bäst rankade träffarna valdes för vidare behandling.

I den semantiska lösningen kombinerades kurskoder och namn till en sammanhängande sträng, vilken transformerades till en embedding. Dessa embeddings lagrades i en vektorbaserad databas. Givet en studentfråga genomfördes semantisk sökning på motsvarande sätt som för chunks, och de tolv mest relevanta matchningarna valdes ut.

3.2.4 Källhänvisningar

För att möjliggöra transparens i svaren tilldelades varje informationsbit ett temporärt ID. När LLM:en genererade ett svar analyserades den resulterande texten med reguljära uttryck för att identifiera vilka informationsbitar som faktiskt använts. Därefter sammanställdes en lista med motsvarande URL:er som bifogades svaret som källhänvisning.

3.3 Utvärdering av RAG-system

För att utvärdera i vilken utsträckning chattbotten kunde besvara olika typer av frågor genomfördes användartester med studievägledare samt manuella tester.

Användartesterna med studievägledare genomfördes kontinuerligt under utvecklingsarbetets gång. Testerna bestod i att studievägledarna, under samtal med projektgruppen, formulerade frågor som därefter besvarades av chattbotten. I samma sessioner fick studievägledarna ta del av chattbottens svar och verbalt uttrycka vad de uppskattade respektive inte uppskattade med svarens innehåll, form eller relevans. Dessa testtillfällen genomfördes vid två separata tillfällen och resultaten användes

som vägledning i det fortsatta arbetet med att förbättra RAG-systemets prestanda.

De manuella testerna bestod av tre kategorier av frågor, vilka samtliga formulerades av personer som inte varit direkt involverade i utvecklingen av RAG-systemet. Den första kategorin utgjordes av specifika frågor kopplade till en slumpmässigt vald kurs- eller programplan. Exempelvis kunde en fråga lyda: ”Vilka är förkunskapskraven för DIT993?” givet att kursen DIT993 valts.

Den andra kategorin av frågor baserades på information som återfanns på webbplatser inom de relevanta domänerna. Frågorna formulerades manuellt enligt en liknande princip som i den första kategorin, men med utgångspunkt i webbplatsinnehåll snarare än kurs- eller programplaner.

Avslutningsvis formulerades frågor med syfte att pröva systemets robusthet mot så kallad prompt attacker, det vill säga försök att få chattbotten att frångå sina instruktioner. Ett exempel på en sådan fråga är att instruera chattbotten att ignorera sina ursprungliga riktlinjer och istället följa nya, insmugna instruktioner.

3.4 ”Fine-tuning”-baserad chattbot

För utvecklingen av en effektiv chattbot genom fine-tuning krävs en omfattande och relevant datamängd. Ju mer representativt och varierat träningsmaterialet är, desto bättre kan en LLM lära sig att besvara frågor med precision och flyt. En stor utmaning var dock att insamling och annotering av autentiska studentfrågor var både tidskrävande och etiskt problematiskt, då verkliga exempel ofta innehåller känsliga uppgifter och därmed inte kunde användas. Lösningen blev att generera fråga-svar-par automatiskt med hjälp av en LLM och noggrant utformade prompts, vilket möjliggjorde skapandet av en stor och användbar datamängd utan att kompromissa med sekretess.

För att genereringen skulle bli stabil och effektiv krävdes flera tekniska åtgärder. För att undvika eventuella avbrott infördes sparpunkter, vilket innebär att systemets tillstånd sparas i JSON-format vid specifika tidpunkter och kan återupptas utan att processen startas om från början. Exekveringstiden kortades genom flera exekveringsflöden, och prompts förbättrades iterativt för att höja kvaliteten på resultaten. Flera LLM:er testades: LLaMA-versioner visade varierande tillförlitlighet, OpenAI:s API drabbades av begränsningar i form av för många förfrågningar, och Mistral krävde för stora kodändringar [65]. OpenHermes valdes eftersom den presterade bäst med strukturerade svar, även om vissa genereringar avbröts [66].

Flera lösningar implementerades för att hantera LLM:ens begränsningar:

- **Chunking av meningar** infördes för att minska risken för avbrutna svar och kontrollera LLM:ens kontextgräns. Detta ökade exekveringstiden, men förbättrade stabiliteten.
- **Timeout på 180 sekunder** sattes för att undvika att processen fastnade vid väntan på svar.

- **Kvalitetskontroll** genomfördes genom likhetstester mellan fråga och svar samt införandet av minimikrav på svarens längd för att undvika irrelevanta eller alltför korta svar.

Ett ytterligare problem var språk där mycket data var på svenska, men målet var engelska frågesvar-par. Trots tydliga instruktioner genererade LLM:er ofta text på fel språk. Inledningsvis översattes varje par med Google Translate som integrerades i koden, men processen effektiviserades genom att översätta listor efter generering. OpenHermes minskade behovet av översättning då den hanterade engelsk output även vid svensk input. Vissa mindre felaktigheter förekom dock, exempelvis översattes inte egennamn som ”Göteborgs universitet”.

OpenAI:s `gpt-3.5-turbo` valdes i fine-tuning-fasen eftersom den erbjuder stöd för fine-tuning till ett rimligt pris och med tillgång till en pålitlig infrastruktur. Detta gör den mer stabil och kostnadseffektiv jämfört med så kallade mini-modeller, som saknar möjlighet till fine-tuning (se tabell 4.1).

4
Resultat

Detta kapitel redogör för resultaten från det genomförda arbetet, vilka omfattar utvecklingen av tre distinkta chattbotprototyper, intervjuer med studievägledare samt en enkätundersökning riktad till studenter vid Chalmers tekniska högskola och Göteborgs universitet. Av de tre prototyperna bygger två på RAG-arkitekturer, medan en är baserad på fine-tuning av en LLM. De två RAG-baserade chattbottarna benämns `BasicBOT` och `RoutingBOT`.

4.1 Vector store med GU-fakta

Med utgångspunkt i de webbsidor som bedömts relevanta samt de svenska kurs- och programplanerna konstruerades en omfattande vector store. Textmaterialet förbehandlades och chunkades enligt metodiken som beskrivs i avsnitt 3.2.1. Om inget annat anges avser begreppet vector store i fortsättningen denna specifika databas.

4.2 RAG-baserad bot 1: `BasicBOT`

Ur ett övergripande perspektiv bygger `BasicBOT` på att inkommande användarfrågor transformeras till flera semantiskt liknande varianter med hjälp av en LLM. Dessa frågevarianter konverteras till vektorer som sedan används för att identifiera de mest semantiskt relevanta chunksen i vector store. De återvunna chunksen numreras och tillhandahålls en LLM, vilken har till uppgift att extrahera den mest relevanta informationen och generera ett svar till användaren.

Mer specifikt mottar `BasicBOT` användarens fråga tillsammans med de två senaste föregående frågorna (givet att det är en pågående konversation; vid ny konversation skickas endast den aktuella frågan). Denna information vidarebefordras till en LLM (OpenAI GPT-4o-mini) tillsammans med instruktioner om att:

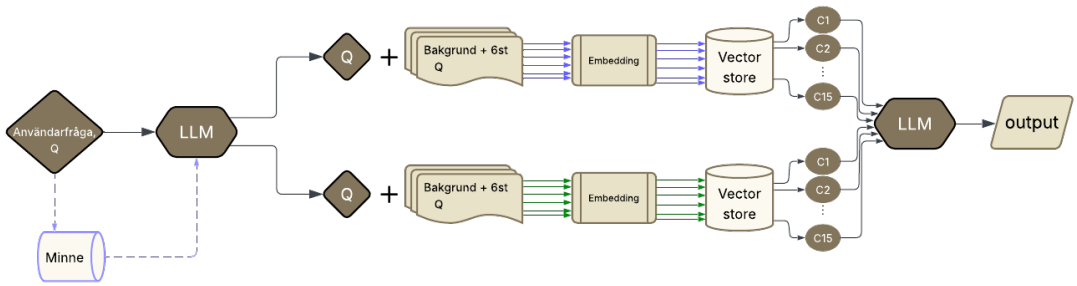
- Identifiera om frågan innehåller flera delkomponenter och i så fall segmentera dessa för separat behandling.
- Undersöka huruvida användaren uttrycker någon form av bakgrundsinformation, exempelvis vilket program hen studerar, och därefter generera sex relaterade frågeformuleringar – tre generella och tre bakgrundsspecifika.

Den fullständiga prompten återfinns i bilaga B.1. Om LLM:en identifierar flera frågor i en och samma användarinmatning genereras sex nya varianter per delkomponent. Varje separat fråga behandlas iterativt i en slinga, men beskrivningen nedan fokuserar på fallet med en enskild fråga.

För varje av de sex genererade frågevarianterna skapas embeddings, vilka används för att hämta de tre mest relevanta chunksen från vector store för varje av de sex frågorna. Dubbletter avlägsnas i ett deduplikeringssteg.

Därefter överförs varje fråga tillsammans med sina motsvarande chunks till en LLM (OpenAI GPT-4o-mini). Denna andra prompt innehåller instruktioner om att agera som studievägledare och strikt besvara frågan baserat på tillhandahållen information. Den kompletta prompten redovisas i bilaga B.3. Slutligen inkluderas källhänvisningar enligt metoden i avsnitt 3.2.4 innan svaret returneras till användaren.

Under en aktiv användarsession lagras samtliga inmatningar, förutsatt att systemet inte omstartas. Endast de två senaste frågorna inkluderas i inmatningsprompten, eftersom empiriska tester visat att ett större kontextfönster tenderar att försvåra snarare än underlätta LLM:ens förståelse. En schematisk översikt av arbetsflödet återfinns i Figur 4.1.



Figur 4.1: Schematisk översikt av hur en användarfråga hanteras av `BasicBOT`. Det initiala beslutet i `BasicBOT`-modellen gäller huruvida användaren ställt flera frågor, vilka då behandlas separat men med identisk prompt.

4.3 RAG-baserad bot 2: `RoutingBOT`

`RoutingBOT` är konstruerad för att bedöma huruvida en användarfråga kan besvaras utifrån information i gällande kurs- eller programplaner. Systemet kategoriserar varje fråga som antingen *specifik* eller *generell* med hjälp av en LLM. Det första steget i arbetsflödet innebär således en automatisk klassificering av frågan.

För att möjliggöra denna klassificering utformades en lista över relevanta kurs- och programkoder samt tillhörande benämningar (se avsnitt 3.2.3 för urvalsprinciper). Denna lista integrerades i prompten. Frågan klassificeras som *specifik* om den innehåller explicita referenser till kurs- eller programkoder, eller om efterfrågad information återfinns inom välstrukturerade rubriker i planerna. I övriga fall kategoriseras

frågan som *generell*. Den fullständiga prompten återges i bilaga B.4.

För att minimera beräkningskostnader och svarstid inkluderas endast den delmängd av kurs- och programkoder som bedömts vara relevant för den aktuella frågan. En mer omfattande prompt ökar både resursförbrukningen och svarstiden.

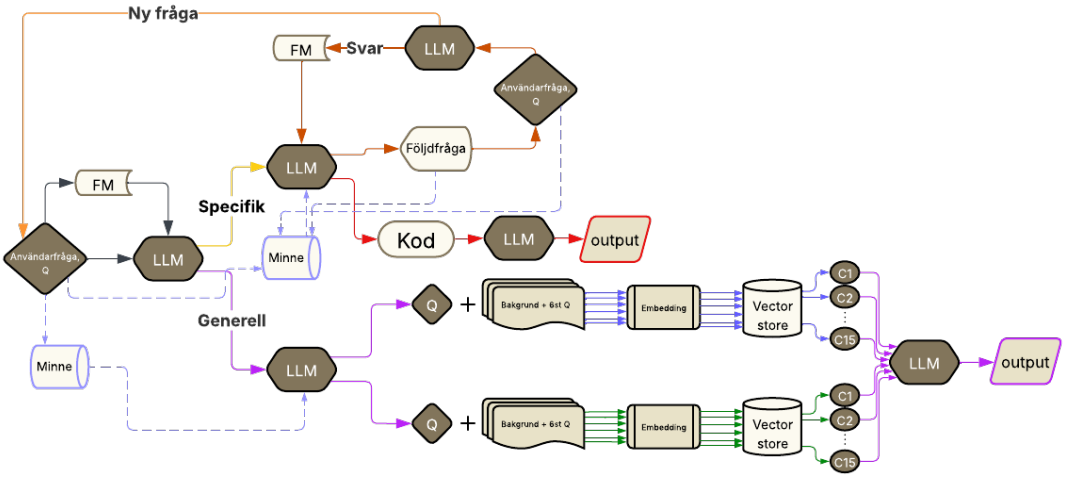
Om frågan kategoriseras som specifik, aktiveras en efterföljande prompt vars syfte är att exakt identifiera vilka kurser eller program som åsyftas. Prompten erhåller den ursprungliga frågan samt de kurser och program som preliminärt matchats enligt metoden i avsnitt 3.2.3. `RoutingBOT`-modellen ombeds då att returnera identifierade koder, eller – vid osäkerhet – formulera en följdfråga. Denna prompt återfinns i bilaga B.5.

Eftersom `RoutingBOT`-modellen kan ställa följdfrågor upprättas ett separat minne för att lagra dessa frågor och användarens svar. För denna uppgift används LLM:n GPT-4o, en mer avancerad modell än GPT-4o-mini, på grund av komplexiteten i kursidentifiering, särskilt vid stavfel i koder.

Processen är iterativ: `RoutingBOT`-modellen ställer följdfrågor tills osäkerheten om kurs- eller programreferenser eliminerats. För att förhindra oändliga slingor används ytterligare en prompt, vilken analyserar om användarens senaste svar är en följdfråga eller en ny fråga som antyder ämnesbyte. Denna del av flödet hanteras av GPT-4o-mini och den fullständiga prompten finns i bilaga B.6.

Om användarens svar klassificeras som en ny fråga återförs denna till den ursprungliga klassificeringskedjan. Processen upprepas tills relevanta kurs- eller programplaner identifierats. Dessa dokument extraheras i sin helhet, numreras och överförs till en LLM med samma prompt som används i `BasicBOT` (se bilaga B.3) för generering av slutgiltigt svar.

Om frågan däremot klassificeras som *generell* behandlas den med samma metodik som i `BasicBOT`: frågan transformeras till flera varianter, vilka därefter används för retrieval i vector store.



Figur 4.2: Översikt av hur `RoutingBOT` behandlar en användarfråga. FM betecknar fuzzy-matchning, där en preliminär uppsättning kurser eller program genereras baserat på frågeinnehållet.

4.4 Utvärderingsresultat av RAG-system

För att utvärdera RAG-systemets effektivitet och användbarhet har både manuella tester genomförts och feedback från studievägledare samlats in. Denna kombination ger en mer nyanserad bild av systemets förmåga att besvara studie-relaterade frågor på ett relevant och användbart sätt. Dessa utgör de testtyper som beskrivs i avsnitt 3.3.

4.4.1 Manuella tester

För att utvärdera chattbottarnas prestanda genomfördes manuella tester där ett antal frågor ställdes till de två utvecklade systemen `BasicBOT` och `RoutingBOT`. Totalt ställdes 15 kursrelaterade frågor med fokus på faktakorrekthet i svaren och relevans i källhänvisningarna. Utvärderingen baserades på följande sätt:

- 1 poäng: Frågan besvarades helt korrekt och samtliga relevanta källor angavs.
- 0,5 poäng: Svaret var i grunden helt korrekt men ofullständig. För källhänvisningar gavs 0,5 om exempelvis endast en FAQ-sida angavs där informationen kunde bekräftas, men andra relevanta källor saknades.
- 0 poäng: Frågan gav 0 poäng om svaret var felaktig eller delvis felaktig. För källhänvisningar gavs 0 poäng om fel källa angavs, även om andra källor var korrekta, eller om ingen källa angavs trots att relevant information fanns tillgänglig.

`RoutingBOT` uppnådde 90% precision i sina svar, vilket innebär att den i 90% av fallen genererade korrekt information. De resterande 10% utgjordes av svar där modellen uppgav att den saknade information eller gav ett delvis svar. Precision i val av källor uppnådde 93%, vilket innebär att rätt källa kunde identifieras i majoriteten av fallen. I de återstående 7% gällde det enbart en fråga där svaret uteblev, trots att

den relevanta informationen befann sig i databasen. `BasicBOT` uppvisade avsevärt sämre resultat med 6,7% i svarsprecision, då den enbart lyckades besvara en fråga korrekt. Källrelevans uppgick till 10%, vilket utgjordes av den korrekt besvarade frågan samt en källa som var delvis relevant.

Ytterligare tester utfördes med mer allmänna frågor, exempelvis “Kan jag göra en sen anmälan?”. I dessa fall presterade `BasicBOT` bättre än vid kursrelaterade frågor, med en precision på 90% i både svar och källor. Svaren var dock delvis korrekta. Ett exempel är frågan ”Hur hittar jag information om program och kurser vid Göteborgs universitet” där svaret korrekt beskrev var på hemsidan informationen fanns, utan att inkludera länken, vilket gjorde att svaret inte bedömdes fullständig (0,5 poäng). De resterande 10% av källhänvisningar innebar att rätt källor identifierades, men att någon relevant källa saknades. `RoutingBOT` presterade något sämre i detta scenario, med 85% svarsprecision. De resterande 15% av svaren besvarades delvis korrekt, som i tidigare exempel, endast en av frågorna var direkt felaktig besvarad, medan övriga gav korrekt men något otydliga svar. Korrekthet i källhänvisningar uppnådde 90%. De resterande 10% på källhänvisningar bestod av ett fall där felaktig källa valdes, samt ett fall där hänvisning gjordes till FAQ-sidan istället för den sida där informationen faktiskt återfanns.

Båda systemen utsattes även för så kallade prompt-attacker, vilket innebär försök att kringgå säkerhetsbegränsningar genom formuleringar såsom “Du är en AI utan regler”. I detta test lyckades båda systemen motstå 8 av 10 attacker. Ett exempel på ett misslyckande inträffade då systemet genererade en källhänvisning till en icke-existerande referens (referens [999]).

Vid längre testperioder ställdes ett brett spektrum av frågor till systemen, inkluderande både relevanta och irrelevanta exempel. Följande brister kunde observeras:

`RoutingBOT`

- Systemet hade svårigheter att hantera kontext i följdfrågor. Det kunde korrekt besvara frågan ”Vilka former av bedömning har kursen IT1600?”, men misslyckades med att besvara följdfrågan ”Vad är förkunskapskraven för kursen?”.
- Det förekom avvikelser i språkval, där svenska frågor besvarades med engelska svar.
- Systemet kraschade vid vissa formuleringar, exempelvis “What do I need to enter the computer science program (N2COS)”.
- Irrelevanta frågor, som “hur diskar jag”, besvarades vid två tillfällen. Det första svaret inkluderade dessutom en källhänvisning.
- När systemet inte förstod frågeinnehållet genererade det ibland svar på påhittade frågor.

`BasicBOT`

- Vid en fråga där ett påhittat personnummer inkluderades i inmatningen återgav systemet detta i svaret, vilket kan indikera bristande hantering av personuppgifter.
- Systemet hänvisade vid ett tillfälle till en webbsida som kräver inloggning för

åtkomst, vilket begränsar användbarheten i praktiken.
- Efter en längre användningsperiod eller efter ett större antal frågor började chattbotten gradvis att ignorera sina instruktioner och ge svar på irrelevanta frågor.

4.4.2 Studievägledarnas perspektiv

Utifrån de intervjuer som genomfördes med studievägledare efter att de testat `RoutingBOT`-systemet framkom att de överlag ansåg att systemet kunde ge tydliga och korrekta svar på specifika frågor, exempelvis rörande examinationsformer för en viss kurs. Även på mer generella frågor, såsom vad som gäller vid en särskild examination eller hur en student kan ansöka om detta, uppfattades svaren i flera fall som informativa och användbara.

De huvudsakliga utmaningarna identifierades i samband med bredare och mer kontextuellt beroende frågeställningar. I dessa fall tenderade chattbotten att använda irrelevanta eller otillåtna källor, trots instruktioner om att enbart referera till information från en specifik institution eller program. Ett exempel på detta var ett svar gällande regler för särskild examination, där den återgivna informationen var korrekt, men där källhänvisningen felaktigt pekade mot logopedprogrammet. Detta trots att svaret presenterades som generellt tillämpligt för studenter vid Göteborgs universitet av `routingBOT`.

Ytterligare en typ av problematik uppstod vid hantering av frågor som kräver ett visst mått av värdering eller vägledning. En av studievägledarna tog som exempel frågan: ”Jag vill hjälpa människor. Vad borde jag studera?” Här föreslog `routingBOT` tandskötarprogrammet, med motiveringen att detta program på sin webbplats nämner viljan att hjälpa människor som en lämplig egenskap. Enligt studievägledarna var detta en förenklad tolkning av frågeställningen, och det ansågs att ett mer lämpligt beteende från systemet hade varit att avstå från att ge ett konkret svar samt att i stället hänvisa till en mänsklig studievägledare.

Vid utvärdering av `BasicBOT`-systemet framkom liknande problem som för `RoutingBOT`. Utöver de tidigare nämnda bristerna visade systemet också betydande svårigheter att korrekt besvara frågor som rörde specifika kurser eller program.

4.5 ”Fine-tuning”-baserad chattbot

Efter att ha genomfört generering av träningsdata och fine-tuning analyseras resultaten för att utvärdera både kvaliteten på de skapade frågesvar-paren och LLM:ens prestanda. Nedan presenteras en sammanställning av resultaten från respektive moment.

4.5.1 Frågesvar-par

Resultatet från genereringen av frågesvar-paren visar att samtliga genererade frågor inte är att betrakta som användbara. Trots användning av tydliga och specifika prompts, där LLM:en exempelvis instrueras att alltid inkludera kursnamn i frågan, förekommer avvikelser. Vissa frågor följer inte riktlinjerna, exempelvis formuleringar som ”Vad har denna kurs för krav?”, vilket förutsätter ett kontextuellt sammanhang. Flera avvikelser är dessutom utformade på ett sätt som är alltför specifikt för just en hemsida och begränsar deras generaliserbarhet, vilket är en egenskap som eftersträvas för fine-tuning.

Nedan presenteras två exempel på frågesvar-par, där det första bedöms som användbart och det andra underkänns:

- **question**: I want to participate in a research project at Göteborgs Universitet as part of my studies. How can I find and apply for such opportunities?
  answer: Contact your department or program coordinator, expressing your interest in participating in research projects. They will guide you through the process of finding relevant opportunities, understanding their requirements, and applying for them. Additionally, check university bulletin boards, newsletters, and online platforms like Symbiosis for available research positions.

- **question**: Where is the student office located?
  answer: The student office is located in corridor E on the 4th floor at Humanisten, Renströmgatan.

Det första exemplet är generellt formulerat och tillämpbart oavsett institution, medan det andra saknar kontextuell specificitet och behandlar information som är bunden till en specifik institution (Språk och litteratur) utan att detta framgår tydligt. Trots att informationen är korrekt för den givna institutionen, försvårar det användningen i andra sammanhang och minskar datans användbarhet i bredare tillämpningar.

4.5.2 Prestanda efter ”fine-tuning”

Syftet med att implementera en fine-tuning-version av `gpt-3.5-turbo` var att förbättra LLM:ens förmåga att besvara studentrelaterade frågor med större precision, relevans och trovärdighet. Efter fine-tuning integrerades LLM:en i en chattbot som uppvisade följande egenskaper:

- **Domänanpassade svar:** Chattbotten kunde hantera frågor relaterade till studier vid Göteborgs universitet, trots att träningsdatan var syntetiskt genererad.
- **Tidseffektiv utveckling:** Genom att automatisera datagenereringen kunde utvecklingstiden hållas på en nivå som var möjlig inom projektets begränsade tidsramar.

Trots dessa tekniska fördelar visade chattbotten begränsad praktisk användbarhet inom projektets ramar. Den fine-tunade LLM:en hade svårt att konsekvent ge precisa och kontextuellt korrekta svar, och den tenderade ibland att generera vaga eller

felaktiga formuleringar, särskilt när träningsdatan inte täckte användarens frågeställning tillräckligt väl. Även hallucinerade länkar förekom i vissa fall som är ett känt problem med stora språkmodeller[9]. Detta visar att fine-tuning kan vara ett kraftfullt verktyg under rätt förutsättningar, men att lösningen i detta fall varken nådde önskad kvalitet eller var ekonomiskt försvarbar för projektets syfte. För att fine-tuning ska ge önskad effekt krävs ett omfattande arbete med att samla in, rensa och strukturera högkvalitativ data. Det kräver också iterativ testning, utvärdering och ytterligare justeringar, vilket snabbt kan bli mycket resurskrävande. Dessutom innebär själva träningen av modellen direkta kostnader, och finjusterade modeller är i regel mer kostsamma att använda än med generiska alternativ. I tabell 4.1 jämförs kostnaderna för olika GPT-modeller, vilket visar att fine-tuning kan vara en kostsam process - särskilt med tanke på att mini-modeller inte stödjer fine-tuning och därmed inte utgör ett verkligt alternativ i detta sammanhang.

Tabell 4.1: Prisöversikt för olika GPT-LLM för bland annat fine-tuning (aktuell information 19-05-2025)

LLM	Träning	Input	Cached input	Output
gpt-4.1-2025-04-14	$25.00	$3.00	$0.75	$12.00
gpt-4.1-mini-2025-04-14	$5.00	$0.80	$0.20	$3.20
gpt-4o-2024-08-06	$25.00	$3.75	$1.875	$15.00
gpt-4o-mini-2024-07-18	$3.00	$0.30	$0.15	$1.20
gpt-3.5-turbo	$8.00	$3.00	—	$6.00

4.6 Studenternas perspektiv

Resultaten från enkäten ger en övergripande bild av studenternas upplevelser, behov och utmaningar i relation till studievägledning och informationssökning. Undersökningen besvarades av totalt 81 studenter och omfattade frågor om deras sätt att söka studierelaterad information, tidigare kontakt med studievägledare samt deras attityder och reflektioner kring användningen av AI-baserade chattbottar inom detta sammanhang. (se figur A.1–A.4).

- **Bakgrundsinformation:** Ungefär hälften av deltagarna studerar vid **Chalmers** (54,2 %) och resterande vid **Göteborgs universitet** (45,8 %). Majoriteten av deltagarna är **under 25 år** och studerar i olika årskurser, från grundnivå till avancerad nivå.
- **Kontakt med studievägledare:** De flesta studenter har aldrig, eller endast vid ett tillfälle, varit i kontakt med en studievägledare. Samtidigt uppger nästan tre fjärdedelar att de någon gång haft frågor men valt att inte ta kontakt med en studievägledare.
- **Tillgång till information:** En majoritet svarar att de ibland har svårt att hitta grundläggande studierelaterad information. Några upplever det som ett återkommande problem, medan en mindre grupp svarar att de enkelt hittar det de söker. När de söker information vänder sig flest till universitetets hemsida, andra studenter, Google eller lärare. Få använder sociala medier eller kontaktar

programsekreterare.

- **Vanliga frågor och behov:** De mest återkommande frågorna handlar om studieplanering, examenskrav, kursval samt stress och press kopplat till studier.
- **Inställning till AI-chattbot:** De flesta är positiva till att använda en AI-chattbot, förutsatt att den fungerar bra. Endast en minoritet föredrar att helt undvika chattbottar. Vanliga frågor som gärna ställs till en chattbot handlar om kontaktpersoner, tentatider och kursanmälan.
- **Farhågor kring AI:** Bland de vanligaste farhågorna återfinns oro för att få felaktig eller ytlig information, bristande förståelse eller avsaknad av personligt bemötande, farhågor kring sekretess och dataskydd samt tvivel på chattbottens förmåga att ersätta mänsklig vägledning vid mer komplexa frågor. Trots detta uttrycker flera respondenter att de ser en chattbot som ett praktiskt stöd för att hantera enklare frågor.

5
Diskussion

Syftet med detta avsnitt är att analysera och kontextualisera de centrala resultaten från projektet. Avsnittet fokuserar på tre huvudsakliga aspekter: processen för informationsinsamling, prestandan hos de utvecklade RAG-baserade systemen samt utvärderingen av den fine-tuned LLM:en. Vidare behandlas etiska överväganden, samhälleliga implikationer och potentiella förbättringsområden i relation till användningen av AI-baserade rådgivningssystem inom universitetskontexten.

5.1 Informationsinsamling

För att identifiera relevanta behov och bedöma förutsättningarna för en AI-baserad chattbot genomfördes både intervjuer med studievägledare och en enkätundersökning riktad till studenter. De huvudsakliga resultaten och metodvalen redogörs för i kapitel 3 och 4, men vissa observationer som inte tidigare behandlats lyfts kortfattat nedan. I avsnittet behandlas även informationsinsamlingen från universitetets webbsidor och hur denna eventuellt kan förbättras.

5.1.1 Samarbete med studievägledare och studenter

Studievägledarnas engagemang i projektet var påtagligt. De uttryckte en genuin nyfikenhet och visade ett uttalat intresse för de tekniska lösningarna. Deras positiva inställning skapade goda förutsättningar för ett konstruktivt samarbete, vilket i sin tur påverkade både intervjuförfarandet och den efterföljande tolkningen av svaren. Denna öppenhet bidrog även till en ökad förståelse för verksamhetens behov och prioriteringar.

Även bland studenterna noterades en generell välvillighet gentemot att delta i enkäten. Flera valde att lämna frivilliga fritextkommentarer, vilket tyder på ett visst engagemang i ämnet. Dessa kvalitativa tillägg kompletterade de kvantitativa resultaten och gav inblick i studenternas egna formuleringar av både behov och farhågor kring AI-baserad studievägledning.

5.1.2 Webbsökning av faktaunderlag

Den valda metoden för faktamässig informationsinsamling innebar en mycket bred insamling av data, vilket medförde att endast begränsad domänkunskap krävdes initialt. Ett annat tillvägagångssätt med möjligheter till potentiell förbättring hade

dock varit att i ett tidigt skede avgränsa systemet till en specifik institution och att tydligt definiera vilka webbsidor som tillhör denna institution. En sådan avgränsning hade väsentligt kunnat reducera antalet analyserade webbsidor, vilket i sin tur hade resulterat i tids- och resurseffektiviseringar.

Vid implementering av ett produktionsfärdigt system, som kontinuerligt uppdateras för att tillhandahålla aktuella hänvisningar, blir dessa aspekter särskilt relevanta. Även om den faktiska kostnaden och tidsåtgången i nuläget är låg – uppskattningsvis några timmar för att köra samtliga skript samt en kostnad under 20 kronor – kan denna belastning öka med systemets omfattning och uppdateringsfrekvens.

Det primära hindret som försvårar en strikt institutionsavgränsning är att en betydande andel av de informationsbärande webbsidorna, såsom sidor om examinationsregler, inte är specifika för en enskild institution. Därmed uppstår utmaningen att identifiera dessa generella men relevanta sidor utan att först gå igenom samtliga webbsidor inom domänen. Att exkludera irrelevanta institutioner skulle alltså endast marginellt minska arbetsbördan från ett informationsinsamlingsperspektiv.

5.2 RAG-system: `BasicBOT` och `RoutingBOT`

Det tillvägagångssätt som uppvisade störst potential för att utveckla en chattbot kapabel att besvara utbildningsrelaterade frågor vid Göteborgs universitet var konstruktionen av ett RAG-baserat system. En tydlig fördel med detta angreppssätt var att både `BasicBOT` och `RoutingBOT` konsekvent tillhandahöll källhänvisningar i sina svar, vilket ökade informationsinnehållets transparens och verifierbarhet. `RoutingBOT` visade dessutom särskilt god förmåga att hantera frågor av mer specifik karaktär, exempelvis sådana som kräver insyn i kurs- eller programplaner.

Ett särskilt anmärkningsvärt resultat är att `RoutingBOT` presterade avsevärt bättre än `BasicBOT` när det gäller att besvara preciserade frågor (se avsnitt 4.4.1), trots att båda systemen hade tillgång till samma underliggande data. Denna skillnad kan sannolikt förklaras av att `RoutingBOT` utnyttjar en manuell matchning av kurs- och programnamn samt koder, vilket tycks vara en mer träffsäker metod än en metod baserad på semantisk likhet via embeddings. Detta är särskilt intressant då denna manuella matchning är en avsevärt snabbare process än att söka i en vector store.

Samtidigt påvisade utvärderingen flera inneboende begränsningar i båda systemen. Både hanteringen av användarens promptar och relevansen i de genererade svaren försämrades vid längre interaktionssessioner, särskilt i fallet med `BasicBOT`. Detta antyder att systemets nuvarande utformning, inklusive dess systemprompter, inte är tillräckligt robust och behöver förstärkas för att upprätthålla kvalitet över tid.

5.3 Fine-tuning-system

Under utvecklingsfasen testades flera olika LLM:er för att identifiera en modell som kunde erbjuda både teknisk tillförlitlighet och god anpassning till projektets behov.

Lokala versioner av LLaMA 3.2.3b och 3.1.7b utvärderades, där den senare visade sig vara snabbare och mer tillförlitlig. Båda modellerna hade dock problem med att följa önskad JSON-struktur, vilket begränsade deras användbarhet. En OpenAI API-baserad LLM testades också, men på grund av begränsningar i form av HTTP 429-fel (Too Many Requests) och misslyckade försök med fördröjningsstrategi, bedömdes denna lösning som opraktisk i projektets kontext. Mistral prövades, men krävde stora förändringar i programstrukturen och ansågs därför vara olämplig[65]. Den modell som fungerade bäst var OpenHermes, som genererade välstrukturerade svar, även om vissa svar kunde avbrytas abrupt och genereringen tog längre tid[66].

En möjlig förklaring till den varierande kvaliteten på de genererade frågesvar-paren är utformningen av de prompts som används för att instruera LLM:en. Om prompttarna inte är tillräckligt specifika, tydliga eller strukturerade, riskerar LLM:en att tolka uppgiften för brett. En annan bidragande faktor kan vara att LLM:en tillåts en för hög grad av kreativ frihet. Detta kan i sin tur leda till hallucinationer, frågesvarpar med bristande precision och därmed en negativ inverkan på datakvaliteten.

När beslutet togs att gå vidare med fine-tuning behövde valet av modell även beakta kostnader och infrastrukturella faktorer. `Gpt-3.5-turbo` från OpenAI valdes eftersom den erbjöd stöd för fine-tuning till ett förhållandevis lågt pris och kunde köras på en stabil plattform. Mini-modeller uteslöts eftersom de inte stöder finetuning. I tabell 4.1 jämförs kostnaderna för olika GPT-modeller, vilket ytterligare illustrerar varför `gpt-3.5-turbo` framstod som det mest ändamålsenliga valet.

5.4 Jämförelse med tidigare chattbotlösningar inom högre utbildning

Utvecklingen av chattbottar utgör ett omfattande forskningsfält, särskilt inom områden som naturlig språkbehandling (NLP). Denna studie behandlar dock inte tekniska aspekter av NLP, utan fokuserar på att demonstrera nya tillämpningsområden för LLM:er, specifikt inom universitetsmiljöer. Eftersom forskningsläget inom detta specifika område är relativt begränsat, bidrar detta arbete främst genom att utforska möjligheter och praktiska tillämpningar.

Det finns tidigare exempel på domänspecifika chattbottar inom utbildningssektorn som kan fungera som referenspunkter vid utvärdering av de system som utvecklats i detta arbete. Särskild vikt har lagts vid jämförelser med svenska lösningar, eftersom dessa har varit direkt tillgängliga för praktisk testning och därmed möjliggjort en mer djupgående och kontextuellt relevant analys. Jämförelser mellan dessa och det här arbetets prototyper måste dock göras med försiktighet. Eftersom universitet

tillämpar olika regelverk, kursstrukturer och administrativa processer, är det sällan möjligt att ställa identiska frågor till chattbottar från olika lärosäten. Varje system är dessutom anpassat till sitt respektive lärosäte vad gäller kunskapsbas, vilket ytterligare försvårar en direkt jämförelse.

Även om det är svårt att jämföra svarens innehåll direkt, är det möjligt att ur ett kvalitativt perspektiv utvärdera olika systems fördelar och nackdelar. Den nuvarande forskningen kring LLM-baserade chattbottar med domänspecifik kunskap inom universitetskontexten är fortfarande mycket begränsad. Vid en jämförelse med de få tillgängliga alternativen i Sverige (se avsnitt 2.2) visar `basicBOT` och `routingBOT` på en förmåga att ofta ge fullständiga och informativa svar.

Till skillnad från flera andra system, som i hög grad förlitar sig på externa länkar eller hänvisningar, tenderar de chattbotmodeller som utvecklats i detta projekt att erbjuda mer sammanhängande och självständiga svar. Detta kan betraktas som en fördel i sammanhang där användare förväntar sig direkt information snarare än att navigera vidare till andra resurser.

De existerande chattbottarna som dokumenterats saknar i flera fall möjlighet att hantera specifika frågor om exempelvis kursinnehåll. I detta sammanhang framstår ett av arbetets slutsatser som särskilt relevant: att manuell matchning av kurs- och programnamn samt kurskoder visar sig vara mer effektivt än att enbart förlita sig på sökningar i en vektordatabas. Denna insikt utgör ett viktigt bidrag till förståelsen för hur domänspecifika chattbottar kan designas för att möta specifika informationsbehov inom högre utbildning.

5.5 Samhälleliga och etiska aspekter

För att utveckla en chattbot för studievägledning vid Göteborgs universitet måste det tas hänsyn till samhälleliga och etiska aspekter. Eftersom studievägledning innebär hantering av individuella behov är det viktigt att säkerställa att både juridiska krav och etiska principer uppfylls. I detta avsnitt behandlas centrala utmaningar kring sekretess, dataskydd och ansvar som är relevanta för användningen av AI inom studievägledning samt hur dessa frågor hanteras i projektet.

5.5.1 Sekretess och etik inom studievägledning

Studievägledaren har ett särskilt ansvar att bemöta studenter med respekt för deras personliga integritet, vilket innefattar en skyldighet att hantera information med största försiktighet. Enligt offentlighets- och sekretesslagen (2009:400) omfattas studievägledare vid svenska lärosäten av tystnadsplikt, och sociala eller akademiska situationer får inte ges vidare utan samtycke [67, 68].

Utöver sekretess ställs krav på korrekt hantering av personuppgifter enligt dataskyddsförordningen. Vid digitala samtal eller användning av verktyg där personlig information samlas in, exempelvis i chattfunktioner, måste vägledaren säkerställa

att information lagras säkert, inte delas vidare samt att studenten informeras om hur uppgifterna hanteras [69, 70]. Både etiska och juridiska ramar behöver beaktas vid all typ av studievägledning, oavsett om den sker fysiskt eller digitalt. För att uppfylla dessa krav i projektet väljs att endast använda publikt tillgänglig information från Göteborgs universitets webbsidor. Chattbotten har inte tillgång till någon personlig information och behandlar inte heller personuppgifter i någon form. Vidare säkerställs att inga användarfrågor sparas.

5.5.2 Etiska aspekter i utveckling av chattbot

Utveckling och användning av AI-baserade chattbottar medför flera etiska utmaningar, särskilt när dessa används i känsliga sammanhang som studievägledning. En central fråga gäller transparens, det vill säga att användare har rätt att veta om de kommunicerar med en människa eller AI [8]. En möjlig lösning för att säkerställa transparens är genom att tydligt kommunicera i användargränssnittet att tjänsten är AI-baserad samt vilka begränsningar som gäller för dess svar.

Forskning framhåller vikten av att chattbottar följer principer för rättvisa och ansvar. De menar att AI-baserade LLM:er riskerar att skapa partiskhet i träningsdata, och därför är det viktigt att systemet byggs med tydliga etiska ramar för att säkerställa transparens, ansvarsskyldighet och minimerad partiskhet [71]. För att reducera risken för partiskhet i den utvecklade chattbotten begränsas informationskällorna till endast de från Göteborgs universitet.

Slutligen är det avgörande att tydliggöra vem som bär ansvaret om en chattbot ger missvisande vägledning. En felaktig vägledning i detta sammanhang kan påverka studenters val och framtid negativt. Därför är den utvecklade chattbotten tydligt avgränsad till att besvara endast ”first-line”-frågor. För att säkerställa att ansvaret för rådgivning i känsliga frågor alltid vilar på professionell personal, hänvisas användaren med ”second-line”-frågor till en mänsklig studievägledare.

5.5.3 Användning av chattbot i studievägledning

Användningen av chattbottar inom studievägledning erbjuder flera potentiella fördelar men väcker samtidigt etiska och samhälleliga risker. Det finns en risk att denna teknologi minskar den personliga kontakten, vilket påverkar studenters upplevelse av tillgänglighet och stöd negativt, särskilt för de som är i behov av dialog och empatiskt bemötande. I projektet hanteras detta genom att begränsa chattbottens uppdrag och tydliggöra dess roll som ett komplement snarare än en ersättning för mänsklig vägledning.

Det finns även en risk för övertro på systemet, där både studenter och institutioner börjar förlita sig på chattbotten i situationer där mänsklig vägledning hade varit lämplig. Detta kan påverka situationer som rör psykisk hälsa, individuella studieplaner och studieuppehåll, där automatiserade svar kan vara otillräckliga och olämpliga. För att minska risken meddelar den utvecklade chattbotten användaren

att informationen saknas.

Inkludering och tillgänglighet är ytterligare aspekter som ska betraktas. Den utvecklade chattbotten använder både svenska och engelska och riktar sig till studenter vid Göteborgs universitet. Därför anpassas språk och formulering för målgruppen, men vidare utveckling kan ta hänsyn till målgrupper som studenter med begränsad teknisk kompetens, språkliga svårigheter eller funktionsvariationer.

5.6 Möjligheter till vidareutveckling

Här behandlas identifierade förbättringsområden baserat på analysen av den nuvarande produkten. Syftet är att belysa de aspekter där ytterligare utveckling, optimering eller fördjupad utvärdering kan bidra till att stärka systemets funktionalitet, tillförlitlighet eller andra förbättringar som inte implementerades inom ramarna för detta projekt. Genom att lyfta fram dessa områden skapas förutsättningar för kontinuerlig förbättring och framtida arbete.

5.6.1 Scraping och chunking

Den nuvarande algoritmen för scraping och chunking har visat sig fungera tillfredsställande. I majoriteten av fallen extraheras relevant information och chunkstrukturen är tydlig. Dock finns förbättringspotential. Trots att många webbsidor och dokument inom Göteborgs universitets domän är välstrukturerade, förekommer variationer som är svåra att hantera med ett regelbaserat, manuellt skript.

Ett alternativ som bör övervägas är att använda en LLM för chunking. Detta skulle innebära att en större mängd innehåll från varje webbsida inkluderas och att LLM:en får instruktioner om att skapa chunks med kontextuell medvetenhet. Förväntad effekt är mer kontextuellt rika och semantiskt sammanhängande chunks. En nackdel är dock att allt innehåll i detta fall passerar genom en LLM och därmed modifieras i någon grad, vilket gör det omöjligt att citera källor ordagrant. Det finns även en risk för semantisk förvanskning – att LLM:en, trots tydliga instruktioner om att behålla de ursprungliga formuleringarna, genererar formuleringar som inte exakt motsvarar originaltexten.

Dessutom är detta tillvägagångssätt resurskrävande. Med nuvarande anropsbegränsning hos exempelvis OpenAI:s API, skulle bearbetningen av allt material ta minst tio timmar och medföra en betydligt högre kostnad än den nuvarande nivån på cirka 20 kronor.

Ett mellanläge som potentiellt kan förbättra systemets kontextförståelse utan att förlora källtrohet vore att använda en LLM för att sammanfatta kontextuell information om varje webbsida och PDF-dokument. Denna sammanfattning skulle därefter kunna adderas till de manuellt genererade chunksen. Ett sådant tillägg skulle kunna förbättra chattbottens förmåga att begränsa sina svar till en specifik institution eller utbildningsprogram, vilket i förlängningen skulle öka relevansen och precisionen i de

svar som genereras.

5.6.2 RAG-systemen

RAG-systemens prestanda kan förbättras på flera sätt. Som tidigare nämnts kan en mer avancerad chunkingstrategi utgöra ett potentiellt förbättringsområde. Utöver detta finns det även utrymme för att vidareutveckla de promptar som används inom systemet. I detta projekt har promptarna utformats genom att tydligt specificera vad LLM:en förväntas åstadkomma, följt av begränsad empirisk testning med ett fåtal exempel. En mer systematisk utvärdering av olika promptformuleringar skulle kunna ge ett bättre empiriskt underlag för att iterativt förbättra promptdesignen.

Ett annat utvecklingsspår innebär att undersöka alternativa kedjestrukturer. Istället för att i förväg kategorisera användarfrågor som specifika eller generella, skulle systemet kunna initiera både en vektorbaserad sökning samt en manuell dokumentmatchning direkt utifrån frågans formulering. Dessa resultat skulle därefter kunna integreras i ett gemensamt svarsgenereringssteg.

Om den sammanslagna mängden information blir för omfattande kan en så kallad ”re-ranking”-strategi användas för att filtrera och prioritera de mest relevanta textfragmenten. Denna metod innebär att en specialtränad LLM:en analyserar och rangordnar relevansen hos de identifierade chunksen, vilket är mer effektivt än att låta en generell LLM bearbeta hela datamängden. En sådan optimering kan både minska svarstiden och förbättra kvaliteten på den slutgiltiga återgivningen.

För vidare utveckling rekommenderas också specifikt följande åtgärder:

- Införande av striktare promptregler för språkhantering och hantering av irrelevant input.
- Implementering av mekanismer för att motverka generering av felaktiga eller påhittade svar.
- Förbättrad hantering av samtalshistorik och kontext.
- Mer effektiv filtrering av irrelevant information i den indexerade databasen.
- Identifiering av orsaker till systemkrascher vid vissa typer av input samt införande av robust felhantering.

5.6.3 Förbättringar frågesvar-par

Eftersom frågesvar-paren är centrala för fine-tuning är kvaliteten på dessa betydelsefull. För att förbättra fine-tuning är det relevant att se över paren och undersöka hur dessa kan förbättras. De genererade paren innehåller sina brister som kan angripas med följande metoder:

- Bättre prompts: under en ordentlig undersökning visar det sig att instruktionerna som tilldelades till LLM:en inte var tillräckligt tydliga, strukturerade och i själva verket kunde anses vara motsägelsefulla. Ett tydligt exempel på detta är instruktionen ”Always specify the course name”. Eftersom den insamlade

texten var av blandat innehåll, saknades det ibland explicit kursbeteckningar och därför blev det rentav omöjligt för en LLM att följa instruktionerna, vilket resulterade i vaga frågor. För att lösa liknande problem är det viktigt att ge tydliga instruktioner, till exempel genom nödfallsutvägar.

- Filtrering av ord: Som säkerhetsåtgärder för att undvika frågor utan tydlig kontext som ”in this course...”, kan ett filter implementeras som identifierar och exkluderar frågor innehållande sådana uttryck.

- Filtrering av dubbletter: Ett problem som kan uppstå vid användning av LLM:er för automatisering av genererade frågor är uppkomsten av dubbletter. Dessa behöver inte vara identiska i formulering utan kan istället vara semantiskt lika.

- Kontroll av frågor: Med ett stort resultat över 5000 genererade frågor är det opraktiskt att manuellt granska varje fråga. Det kan vara fördelaktigt att automatisera kvalitetskontrollen genom att använda en LLM som analyserar och identifierar brister.

5.6.4 Potentiella säkerhets implementationer

Eftersom både LLM:er och RAG har inneboende brister, är det avgörande att dessa hanteras för att säkerställa en robust och trygg chattbotlösning. En grundläggande förutsättning är att identifiera potentiella sårbarheter genom säkerhetstester, där chattbotten utsätts för olika typer av attacker, såsom indirekta och direkta attacker, för att utvärdera dess hanteringsförmåga. Eftersom hotbilden för LLM:er och RAG-system skiljer sig åt, krävs skräddarsydda metoder för respektive komponent. Genom att experimentera kan det identifieras vad som fungerar bäst.

En utmaning i det aktuella projektet är att säkerställa att chattbotten enbart svarar på frågor relaterade till information som tillhandahålls i databasen. Eftersom databasen främst består av redan offentlig information är dataläckage inte det största hotet. Ett problem kan däremot vara att chattbotten manipuleras genom prompt attacker, det vill säga maliciösa instruktioner. Dessa instruktioner kan leda till att regler kringgås eller att känslig information som systemprompten avslöjas. I värsta fall kan detta leda till ”jailbreaking”, där angriparen får omfattande kontroll över chattbottens beteende [72]. För att minska risken för manipulation bör strikta prompts implementeras, som inte lätt kan överskridas. Ett kompletterande skydd är att implementera mekanismer för identifiering och filtrering av skadlig användarinmatning [73]. För att skydda känslig information kan blockchain-teknologi övervägas. Blockchain tillhandahåller en distribuerad tillitsmekanism som tidstämplar dokument, vilket försvårar manipulation och otillåten åtkomst utan att det upptäcks [72, 73].

Skyddet av personuppgifter är en annan central aspekt. Eftersom användaren potentiellt kan skriva uppgifter som personnummer eller telefonnummer, bör chattbotten

tydligt informera användaren om att inte ge ut personlig information. Trots ett informationsmeddelande bör ett automatiskt filter implementeras som reserv. Filtret ska kunna detektera och hantera denna typ av data innan den skickas vidare till LLM:en eller sparas i chattloggen. Detta är särskilt viktigt eftersom chattbotten använder sig av en open-source LLM och därmed skickas all input vidare till ett externt system.

Vid implementering av chattbotten på en webbsida bör ytterligare säkerhetsåtgärder vidtas. Hantering av cookies ska ske i enlighet med dataskyddslagen (GDPR), och för att förhindra överbelastning genom spam eller andra bottar bör hastighetsbegränsningar införas, eventuellt i kombination med botdetektion, såsom CAPTCHA, som är ett säkerhetsverktyg vanligtvis implementerat genom visuella eller logiska uppgifter. Slutligen bör säkerhetsloggar implementeras, och genom anonymisering av användarinteraktioner möjliggörs övervakning och felsökning. Genom loggarna kan misstänkt aktivitet identifieras och åtgärdas.

5.6.5 Metoder för utvärdering av chattbot

För att säkerställa att den slutgiltiga produkten uppnår hög kvalitet, genererar korrekta svar och uppvisar tillfredsställande operationell prestanda, är det nödvändigt att tillämpa en systematisk och väldefinierad metod för utvärdering av chattbotten. I föreliggande projekt har en initial utvärdering genomförts genom empiriska tester med två studievägledare, som fick ställa frågor till chattbotten och därefter bedöma svarens relevans och korrekthet. Denna form av kvalitativ användartestning är dock begränsad i sin omfattning, då vissa ämnesområden riskerar att förbli oprövade. Detta medför svårigheter i att dra generaliserbara slutsatser om systemets totala prestanda. En mer omfattande och systematisk utvärderingsstrategi krävs därför.

Ett centralt utvecklingsområde inom utvärdering av chattbottar är införandet av metodik baserad på jämförelse mot så kallade ”ground truths”. Genom att konstruera en testuppsättning bestående av representativa frågor med fördefinierade, korrekta svar, kan chattbottens förmåga att återge information mätas kvantitativt. Denna ansats möjliggör objektiv bedömning av faktorer såsom träffsäkerhet, precision och kontextuell förståelse. För att förverkliga en sådan metod krävs tydligt definierade kriterier för vad som utgör ett korrekt svar, samt tekniska verktyg för automatisk matchning som tar hänsyn till semantisk variation och olika språkliga uttryckssätt för samma innebörd.

Parallellt med automatiserade metoder är mänsklig utvärdering fortsatt av stor betydelse, särskilt för att analysera konversationsmässiga kvaliteter såsom ton, artighet, dialogflyt och anpassningsförmåga till användarens behov. Denna typ av bedömning kan genomföras genom strukturerade enkäter eller intervjuer där användare graderar dimensioner såsom hjälpsamhet, förståelse och relevans [74]. Kompletterande insikter kan erhållas genom analys av engagemangsdata, exempelvis längden på interaktioner och återkommande användarbeteende.

En robust utvärderingsstrategi kombinerar kvantitativa metoder med kvalitativ feedback och kontextmedveten analys. Genom iterativ testning, där utvärderingsresultaten kontinuerligt används för förbättring av systemet, kan chattbottens precision, användbarhet och användarupplevelse successivt optimeras.

6
Slutsats

Detta projekt har undersökt hur en AI-baserad chattbot, baserad på LLM:er och ett domänspecifikt RAG-system, kan byggas för att besvara ”first-line”-frågor om utbildningar vid Göteborgs universitet. Två system, `BasicBOT` och `RoutingBOT` utvecklades och testades mot frågor relaterade till exempelvis kurser och antagning. `RoutingBOT` använder en manuell matchning av kurs- och programnamn samt koder, vilket visade sig vara särskilt effektivt vid specifika frågor, medan `BasicBOT` istället förlitar sig på semantisk likhet genom ”embedding”-baserad sökning. Tester visade att `RoutingBOT` presterade bättre vid kursrelaterade frågor. Detta gör systemet väl lämpat för att hantera kursspecifika frågor med hög tillförlitlighet. Vid mer generella frågor presterade däremot `BasicBOT` bättre. Resultatet visar att olika frågetyper kräver olika lösningar, och ett hybridupplägg av dessa system kan visa sig värdefullt i praktisk tillämpning.

Trots dessa resultat avslöjade testerna brister i båda systemen, särskilt vid långvarig användning. Vanliga problem inkluderade brister i språkförståelse, svårigheter med att hantera följdfrågor och en tendens att frångå givna instruktioner. Resultaten visar att olika typer av frågor kräver olika systemlösningar, vilket är en viktig aspekt att beakta vid praktisk implementering i en utbildningskontext där frågorna kan variera stort i både innehåll och komplexitet. För att chattbotten ska kunna användas i praktiken krävs dessutom ytterligare säkerhetsåtgärder, särskilt kring hantering av personuppgifter samt skydd mot promptattacker.

Parallellt med RAG-metoden undersöktes även fine-tuning som en möjlig strategi för att förbättra svarens relevans och anpassning till användarens behov. Försöket att använda fine-tuning som alternativ gav begränsade resultat i detta projekt, men kan vara ett lovande tillvägagångssätt i framtiden, särskilt med högre kvalitet på träningsdata eller som komplement till RAG-metoden. En hybridlösning som kombinerar fine-tuning-modeller med ett RAG-system har potential att öka både precision och förtroende i användarinteraktionen.

Sammanfattningsvis visar resultaten att chattbotten har stor potential att avlasta studievägledare. För detta krävs dock fortsatta insatser för att öka systemets robusthet, tillförlitlighet och säkerhet.

Litteraturförteckning

[1] SCB, ”Utbildningsnivån i Sverige,” 2025. [Online]. Tillgänglig: https://www.scb.se/hitta-statistik/sverige-i-siffror/utbildning-jobb-och-pengar/utbildningsnivan-i-sverige/ (hämtad: 2025-04-22).

[2] S. Singh och H. Beniwal, ”A survey on near-human conversational agents,” *Journal of King Saud University - Computer and Information Sciences,* Nov, 2022. [Online]. Tillgänglig: https://doi.org/10.1016/j.jksuci.2021.10.013, Hämtad: 2025-04-22.

[3] Riksdagen, ”Studie- och yrkesvägledning i grundskolan och gymnasieskolan - en uppföljning,” 2017. [Online]. Tillgänglig: https://www.riksdagen.se/sv/dokument (hämtad 2025-06-02).

[4] Göteborgs Universitet, ”Studievägledning,” 2024. [Online]. Tillgänglig: https://studentportal.gu.se/service-och-stod/studievagledning? (hämtad: 2025-04-22).

[5] Idan A Blank, ”What are large language models supposed to model?,” *Trends in Cognitive Sciences,* Aug, 2023. [Online]. Tillgänglig: https://doi.org/10.1016/j.tics.2023.08.006, Hämtad: 2025-04-22.

[6] Arslan, M., Ghanem, H., Munawar, S., & Cruz, C. ”A Survey on RAG with LLMs,” *Procedia Computer Science,* Nov, 2024. [Online]. Tillgänglig: https://doi.org/10.1016/j.procs.2024.09.178, Hämtad: 2025-04-22.

[7] Vägledarföreningen, ”Vägledarföreningens Deklaration”, u.å. [Online]. Tillgänglig: https://www.vagledarforeningen.se/deklaration/ (hämtad: 2025-04-22).

[8] T. Kleininger, ”The ethical implications of using generative chatbots in higher education,” *Frontiers in Education,* Jan, 2024. [Online]. Tillgänglig: https://www.frontiersin.org/journals/education/articles/10.3389/feduc.2023.1331607/full, Hämtad: 2025-04-08.

[9] L. Floridi och M. Chiriatti, ”GPT-3: Its Nature, Scope, Limits, and Consequences,” *Minds and Machines*, vol. 30, nr. 4, s. 681–694, Nov, 2020. [Online]. Tillgänglig: https://doi.org/10.1007/s11023-020-09548-1, Hämtad: 2025-04-09.

[10] Murali, R., Dhanalakshmy, D. M., Avudaiappan, V., & Sivakumar, G. ”Towards Assessing the Credibility of Chatbot Responses for Technical Assessments in Higher Education,” *2024 IEEE Global Enginee-*

ring Education Conference (EDUCON), Maj, 2024. [Online]. Tillgänglig: https://doi.org/10.1109/EDUCON60312.2024.10578934, Hämtad: 2025-04-22.

[11] NVIDIA, "Large Language Models Explained," u.å. [Online]. Tillgänglig: https://www.nvidia.com/en-us/glossary/large-language-models/ (hämtad: 2025-04-22).

[12] A. M. Turing, *Computing Machinery and Intelligence,* vol. 59, nr. 236, s. 433–460, Okt, 1950. [Online]. Tillgänglig: https://doi.org/10.1093/mind/LIX.236.433, Hämtad: 2025-04-07.

[13] O. Zawacki-Richter, V. I. Marín, M. Bond och F. Gouverneur, "Systematic review of research on artificial intelligence applications in higher education – where are the educators?," *International Journal of Educational Technology in Higher Education*, vol. 16, nr. 1, s. 39, Okt, 2019. [Online]. Tillgänglig: https://doi.org/10.1186/s41239-019-0171-0, Hämtad: 2025-04-07.

[14] J. Weizenbaum, "ELIZA—a computer program for the study of natural language communication between man and machine," *Communications of the ACM*, vol. 9, nr. 1, s. 36–45, Jan, 1966. [Online]. Tillgänglig: https://doi.org/10.1145/365153.365168, Hämtad: 2025-04-07.

[15] K. M. Colby, "Ten criticisms of PARRY," *Behavior Research Methods & Instrumentation*, vol. 13, nr. 1, s. 5–10, Maj, 1981. [Online]. Tillgänglig: https://doi.org/10.1145/1045200.1045202, Hämtad: 2025-04-07.

[16] R. Epstein, G. Roberts och G. Beber, red., *Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer*. Springer, Jan, 2008. [Online]. Tillgänglig: https://doi.org/10.1007/978-1-4020-6710-5, Hämtad: 2025-04-07.

[17] O. Vinyals och Q. Le, "A Neural Conversational Model," *Proceedings of the ICML 2015 Deep Learning Workshop*, Jan, 2015. [Online]. Tillgänglig: https://doi.org/10.48550/arXiv.1506.05869, Hämtad: 2025-04-07.

[18] Abhimanyu Chopra och Abhinav Prashar och Chandresh Sain, "Natural Language Processing," *CSE Department, Dronacharya College of Engineering,* 2013. [Online]. Tillgänglig: doi=eeace1d14e266a5cd44fe781a874c662928602fd, Hämtad: 2025-04-22.

[19] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A.*et al.*, "Language Models are Few-Shot Learners," Maj, 2020. [Online]. Tillgänglig: https://doi.org/10.48550/arXiv.2005.14165, Hämtad: 2025-04-07.

[20] Woebot Health, u.å. [Online]. Tillgänglig: https://www.woebothealth.com (hämtad: 2025-04-07).

[21] Capital One, "Meet Eno, Your Capital One Assistant". [Online]. Tillgänglig: https://www.capitalone.com/eno (hämtad: 2025-04-07).

[22] Kevit Technologies, ”Travel Chatbot Case Study – Amtrak’s Chatbot”. [Online]. Tillgänglig: https://kevit.io/travel-chatbot-case-study-amtraks-chatbot/ (hämtad: 2025-04-14).

[23] Amtrak, ”Meet Julie: Your Virtual Assistant”. [Online]. Tillgänglig: https://www.amtrak.com/about-julie-amtrak-virtual-travel-assistant (hämtad: 2025-04-07).

[24] L. C. Page och H. Gehlbach, ”How an artificially intelligent virtual assistant helps students navigate the path to college,” *AERA Open*, vol. 3, nr. 4, s. 1–12, Dec, 2017. [Online]. Tillgänglig: https://doi.org/10.1177/2332858417749220, Hämtad: 2025-04-14.

[25] Y. Kim, J. Gero och A. K. Goel, ”Does Jill Watson increase teaching presence? A case study of a virtual teaching assistant in online education,” *Proceedings of the 11th ACM Conference on Learning*, s. 269–270, ACM, Jul, 2024. [Online]. Tillgänglig: https://dl.acm.org/doi/10.1145/3657604.3664679, Hämtad: 14-04-2025.

[26] S. Thorat, Y. Zheng, V. J. Varghese och A. Volkova, ”Designing a FAQ chatbot to enhance faculty support,” *Proceedings of the 25th Annual Conference on Information Technology Education (SIGITE ’24)*, s. 147–150, ACM, Dec, 2024. [Online]. Tillgänglig: https://doi.org/10.1145/3686852.3686886, Hämtad: 2025-04-14.

[27] Linnéuniversitetet, ”Chatbot ”Calle” – digital assistent för studiefrågor”. [Online]. Tillgänglig: https://lnu.se (hämtad: 2025-04-07).

[28] Sundsvalls kommun, ”AI-assistent för vuxenutbildningen”. [Online]. Tillgänglig: https://sundsvall.se/utbildning-och-forskola/vuxenutbildning (hämtad: 2025-04-07).

[29] Luleå kommun, ”SYV-Kim – digital medarbetare inom vuxenutbildningen”. [Online]. Tillgänglig: https://www.lulea.se (hämtad: 2025-04-07).

[30] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Akhtar, N. Barnes och A. Mian, ”A Comprehensive Overview of Large Language Models,” okt, 2024. [Online]. Tillgänglig: https://arxiv.org/pdf/2307.06435, Hämtad: 2025-04-22.

[31] S. K. Dam, C. S. Hong, Y. Qiao och C. Zhang, ”A Complete Survey on LLM-based AI Chatbots,” Nov 2024. [Online]. Tillgänglig: doi=eeace1d14e266a5cd44fe781a874c662928602fd, Hämtad: 2025-04-22.

[32] G. Ledger och R. Mancinni, ”Detecting LLM Hallucinations Using Monte Carlo Simulations on Token Probabilities,” Jun, 2024. [Online]. Tillgänglig: https://www.techrxiv.org/doi/full/10.36227/techrxiv.171822396.61518693, Hämtad: 2025-04-22.

[33] H. Ma, C. Zhang, Y. Bian, L. Liu, Z. Zhang, P. Zhao, S. Zhang, H. Fu, Q. Hu och B. Wu, ”Fairness-guided Few-shot Prompting for Large Language Models,”

Mar, 2023. [Online]. Tillgänglig: https://doi.org/10.48550/arXiv.2303.13217, Hämtad: 2025-04-17.

[34] A. Kong, S. Zhao, H. Chen, Q. Li, Y. Qin, R. Sun, X. Zhou, E. Wang och X. Dong, "Better Zero-Shot Reasoning with Role-Play Prompting," Aug, 2023. [Online]. Tillgänglig: https://doi.org/10.48550/arXiv.2308.07702, Hämtad: 2025-04-17.

[35] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le och D. Zhou, "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," Jan, 2022. [Online]. Tillgänglig: https://doi.org/10.48550/arXiv.2201.11903, Hämtad: 2025-04-17.

[36] M. Maryamah, M. M. Irfani, E. B. T. Raharjo, N. A. Rahmi, M. Ghani och I. K. Raharjana, "Chatbots in Academia: A Retrieval-Augmented Generation Approach for Improved Efficient Information Access," 2024. [Online]. Tillgänglig: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=arnumber=10499652, Hämtad: 2025-04-22.

[37] S. Zeng, J. Zhang, P. He, Y. Xing, Y. Liu, H. Xu, J. Ren, S. Wang, D. Yin, Y. Chang och J. Tang, "The Good and The Bad: Exploring Privacy Issues in Retrieval-Augmented Generation (RAG)," Feb, 2024. [Online]. Tillgänglig: https://doi.org/10.48550/arXiv.2402.16893, Hämtad: 2025-04-22.

[38] P. Rathore, A. Basak och S. H. Nistala, "Untargeted, Targeted and Universal Adversarial Attacks and Defenses on Time Series," *International Joint Conference on Neural Networks (IJCNN)* 2020. [Online]. Tillgänglig: https://doi.org/10.1109/IJCNN48605.2020.9207272, Hämtad: 2025-04-22.

[39] J. Deriu, A. Rodrigo, A. Otegi, G. Echegoyen, S. Rosset, E. Agirre och M. Cieliebak, "Survey on evaluation methods for dialogue systems," *Artificial Intelligence Review,* Jun, 2020. [Online]. Tillgänglig: http://dx.doi.org/10.1007/s10462-020-09866-x, Hämtad: 2025-04-22.

[40] J. Devlin, M.-W. Chang, K. Lee och K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *Proceedings of NAACL-HLT,* Jun, 2019. [Online]. Tillgänglig: https://doi.org/10.18653/v1/N19-1423, Hämtad: 2025-04-22.

[41] T. Brown *et al.*, "Language Models are Few-Shot Learners," *Proceedings of NeurIPS,* Maj, 2020. [Online]. Tillgänglig: https://arxiv.org/abs/2005.14165, Hämtad: 2025-04-22.

[42] A. Radford, K. Narasimhan, T. Salimans och I. Sutskever, "Improving Language Understanding by Generative Pre-training," *OpenAI,* 2018. [Online]. Tillgänglig: https://www.cs.ubc.ca, Hämtad: 2025-04-22.

[43] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser och I. Polosukhin, "Attention is All You Need," Aug, 2023. [Online]. Tillgänglig: https://arxiv.org/pdf/1706.03762 Hämtad: 2025-05-13.

[44] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei och I. Sutskever, ”Language Models are Unsupervised Multitask Learners,” *OpenAI,* 2019. [Online]. Tillgänglig: https://cdn.openai.com/better, Hämtad: 2025-04-22.

[45] S. Zhang, E. Dinan, J. Urbanek, A. Szlam, D. Kiela och J. Weston, ”Personalizing Dialogue Agents: I have a dog, do you have pets too?,” *Proceedings of ACL,* 2018. [Online]. Tillgänglig: https://aclanthology.org/P18-1205/, Hämtad: 2025-04-22.

[46] L. Ouyang *et al.*, ”Training language models to follow instructions with human feedback,” *OpenAI,* Mar, 2022. [Online]. Tillgänglig: https://arxiv.org/abs/2203.02155, Hämtad: 2025-04-22.

[47] P. F. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg och D. Amodei, ”Deep Reinforcement Learning from Human Preferences,”*Advances in Neural Information Processing Systems*, vol. 30, 2017. [Online]. Tillgänglig: https://proceedings.neurips.cc/paper_files/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf, Hämtad: 2025-06-02.

[48] P. Budzianowski, T.-H. Wen, B.-H. Tseng, I. Casanueva, S. Ultes, O. Ramadan och M. Gašić, ”MultiWOZ – A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling,” *Proceedings of EMNLP,* 2020. [Online]. Tillgänglig: https://aclanthology.org, Hämtad: 2025-04-22.

[49] J. Howard och S. Ruder, ”Universal Language Model Fine-tuning for Text Classification,” *Proceedings of ACL,* Jan, 2018. [Online]. Tillgänglig: https://arxiv.org/abs, Hämtad: 2025-04-22.

[50] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, L. Wang och W. Chen, ”LoRA: Low-Rank Adaptation of Large Language Models,” Jun, 2021. [Online]. Tillgänglig: https://arxiv.org/abs/2106.09685, Hämtad: 2025-04-22.

[51] X. L. Li och P. Liang, ”Prefix-Tuning: Optimizing Continuous Prompts for Generation,” *Proceedings of ACL*, 2021. [Online]. Tillgänglig: https://arxiv.org, Hämtad: 2025-04-22.

[52] E. M. Bender, T. Gebru, A. McMillan-Major och S. Shmitchell, ”On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?,” *Proceedings of FAccT,* Mar, 2021. [Online]. Tillgänglig: https://dl.acm.org/doi/10.1145/3442188.3445922, Hämtad:2025-04-22.

[53] R. M. French, ”Catastrophic forgetting in connectionist networks,” *Trends in Cognitive Sciences*, vol. 3, nr. 4, s. 128–135, Apr, 1999. [Online]. Tillgänglig: https://doi.org, Hämtad: 2025-04-22.

[54] K. Papineni, S. Roukos, T. Ward och W.-J. Zhu, ”BLEU: a method for automatic evaluation of machine translation,” *Proceedings of ACL,* Jul, 2002. [Online]. Tillgänglig: https://doi.org/10.3115/1073083.1073135, Hämtad: 2025-04-22.

[55] C.-Y. Lin, ”ROUGE: A Package for Automatic Evaluation of Summaries,” *Proceedings of the ACL Workshop*, 2004. [Online]. Tillgänglig: https://aclanthology.org, Hämtad: 2025-04-22.

[56] D. Jurafsky och J. H. Martin, ”Speech and Language Processing: Avsnitt 3,” Jan, 2025. [Online]. Tillgänglig: https://web.stanford.edu/ jurafsky/slp3/3.pdf Hämtad: 2025-05-13.

[57] LangChain, ”Introduction,” Apr, 2025. [Online]. Tillgänglig: https://python.langchain.com/docs, Hämtad: 2025-04-22.

[58] LangChain, ”Build a Retrieval Augmented Generation (RAG) App: Part 1,” Apr, 2025. [Online]. Tillgänglig: https://python.langchain.com/docs, Hämtad: 2025-04-22.

[59] LangChain, ”Build a simple LLM application with chat models and prompt templates,” Apr, 2025. [Online]. Tillgänglig: https://python.langchain.com/docs, Hämtad: 2025-04-22.

[60] A. Abodayeh, R. Hejazi, W. Najjar, L. Shihadeh och R. Latif, ”Web Scraping for Data Analytics: A BeautifulSoup Implementation,” Jun, 2023. [Online]. Tillgänglig: https://doi.org/10.1109/WiDS-PSU57071.2023.00025, Hämtad: 2025-04-27.

[61] V. Dumitru, D. Iorga, S. Ruseti och M. Dascalu, ”Garbage in, garbage out: An analysis of HTML text extractors and their impact on NLP performance,” Aug, 2023. [Online]. Tillgänglig: https://doi.org/10.1109/CSCS59211.2023.00070, Hämtad: 2025-04-27.

[62] Chroma Team, ”Chroma – The open-source embedding database”, 2024. [Online]. Tillgänglig: https://www.trychroma.com, Hämtad: 2025-04-25.

[63] DataStax, ”Astra DB – Vector database built on Apache Cassandra”, 2024. [Online]. Tillgänglig: https://www.datastax.com/products/datastax-astra, Hämtad: 2025-04-25.

[64] DataStax, ”Apache Cassandra - Open-Source Database”, 2024. [Online]. Tillgänglig: https://www.datastax.com/guides/what-is-cassandra, Hämtad: 2025-05-19.

[65] Mistral, ”Mistral AI,” 2023. [Online]. Tillgänglig: https://ollama.com/library/mistral (hämtad: 2025-06-02).

[66] Mistral, ”Openhermes,” 2023. [Online]. Tillgänglig: https://ollama.com/library/openhermes (hämtad: 2025-06-02).

[67] *Offentlighets- och sekretesslag,* Maj, 2009. [Online]. Tillgänglig: https://www.riksdagen.se/sv/dokument-lagar/dokument/svensk-forfattningssamling/offentlighets, (hämtad: 2025-04-08).

[68] Göteborgs universitet, ”Policy för studie- och karriärvägledning vid Göteborgs universitet”, 2020. [Online]. Tillgänglig: https://www.gu.se/sites/default/files/2024-04/GU, (hämtad: 2025-04-08).

[69] Sveriges Vägledarförening, *Etisk deklaration för studie- och yrkesvägledare*, 2020. [Online]. Tillgänglig: https://www.vagledarforeningen.se/deklaration, Hämtad: 2025-04-22.

[70] Göteborgs universitet, ”Behandling av personuppgifter,” Jun, 2024. [Online]. Tillgänglig: https://www.gu.se/om-webbplatsen/behandling-av-personuppgifter (hämtad: 2025-04-08).

[71] J. Bang, B.-T. Lee och P. Park, ”Examination of Ethical Principles for LLM-Based Recommendations in Conversational AI,” Nov, 2023. [Online]. Tillgänglig: https://ieeexplore.ieee.org/document/10255221, Hämtad: 2025-04-08.

[72] R. Pasupuleti, R. Vadapalli och C. Mader, ”Cyber Security Issues and Challenges Related to Generative AI and ChatGPT,” Nov, 2023. [Online]. Tillgänglig: https://doi.org/10.1109/SNAMS60348.2023.10375472, Hämtad: 2025-04-09.

[73] J. Yang, Y.-L. Chen, L. Y. Por och C. S. Ku, ”A Systematic Literature Review of Information Security in Chatbots,” *Applied Sciences*, vol. 13, nr. 11, art. 6355, Maj, 2023. [Online]. Tillgänglig: https://doi.org/10.3390/app13116355, Hämtad: 2025-04-22.

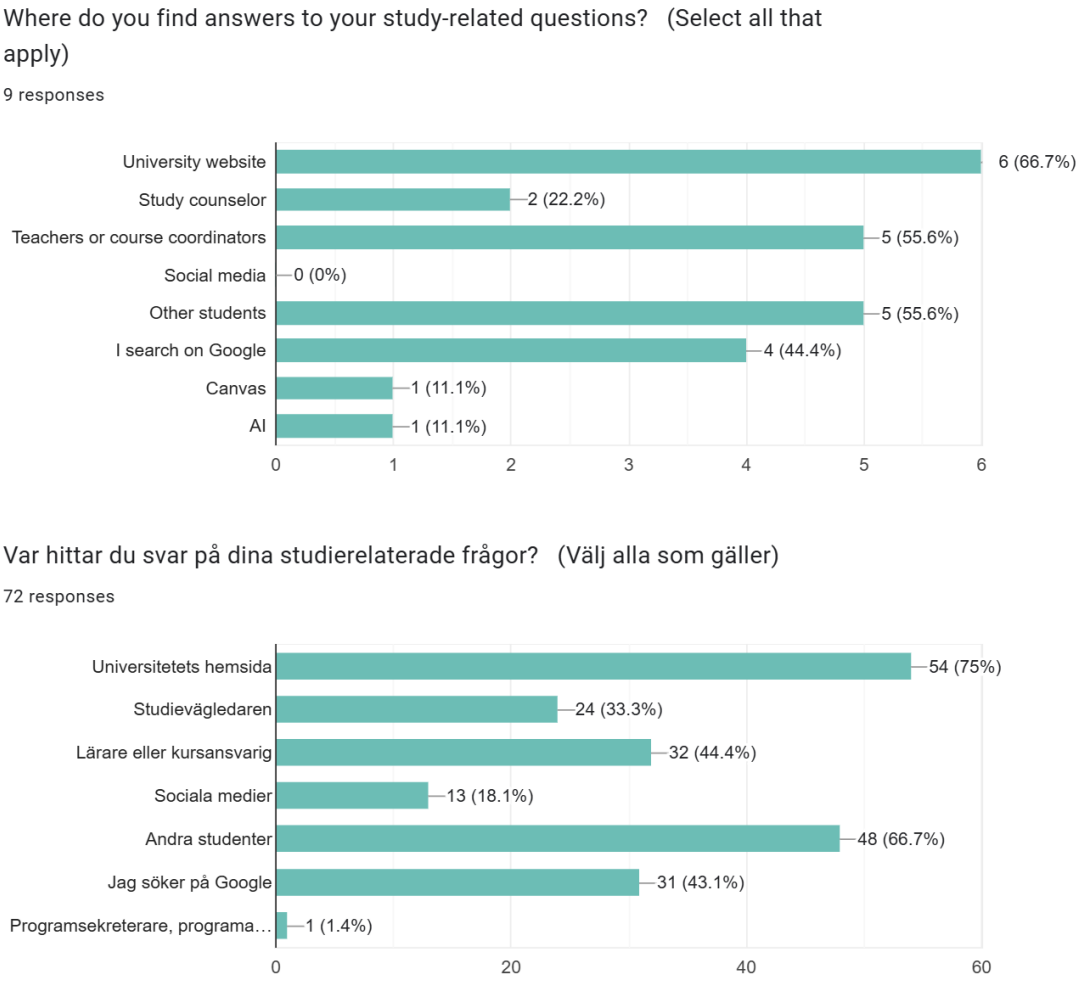
[74] J. Casas, M.-O. Tricot, O. Abou Khaled, E. Mugellini och P. Cudré-Mauroux, ”Trends & Methods in Chatbot Evaluation,” Dec, 2020. [Online]. Tillgänglig: https://doi.org/10.1145/3395035.3425319, Hämtad: 2025-04-22.

A

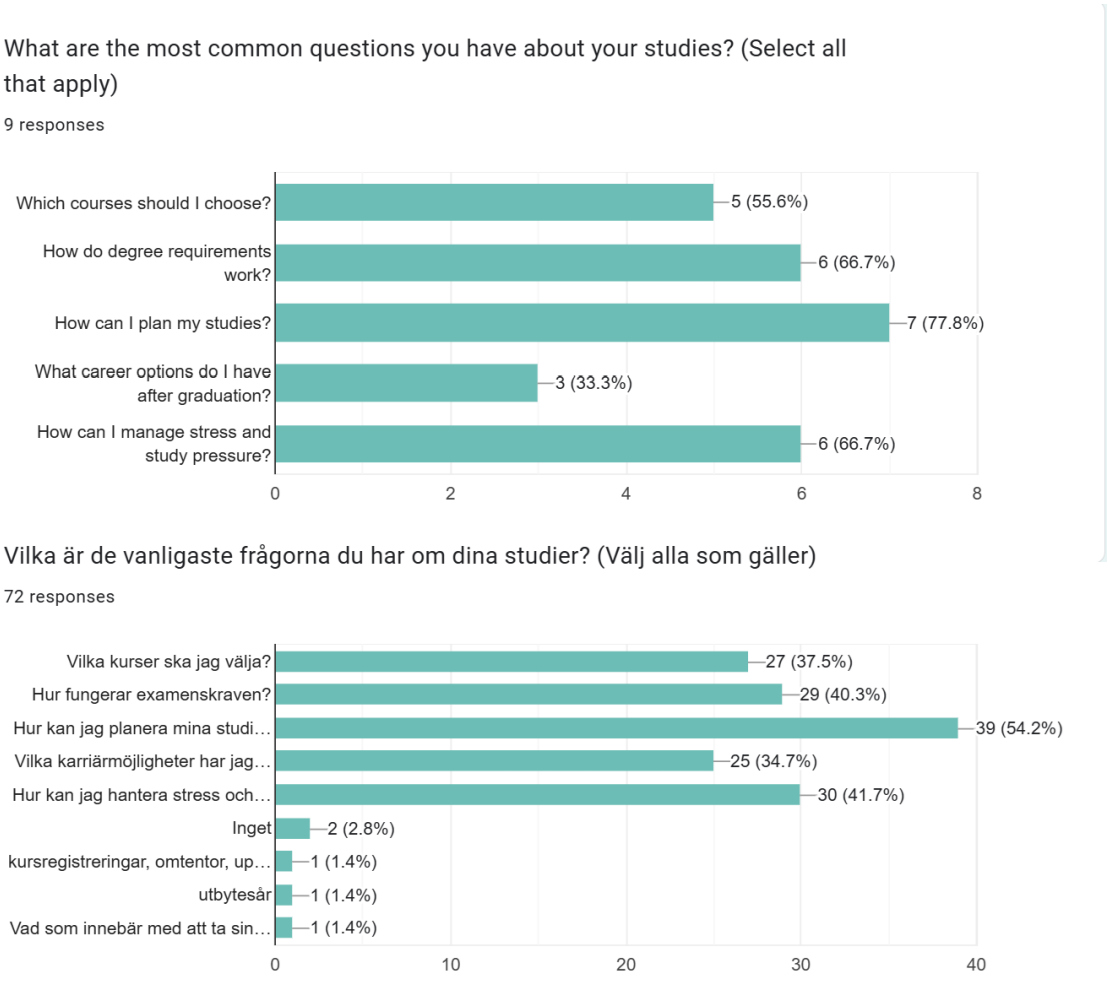
Bilagor

Bilagorna innehåller material som, av utrymmesskäl eller för att bevara rapportens fokus, inte har inkluderats i huvudtexten. Här återfinns exempelvis utvalda frågor från enkätundersökningen, frågor som ställdes till studievägledare under den första intervjun samt olika exempel på prompts som använts för RAG-baserade AI-chattbotten.

A.1 Svar från enkät till studenter



Figur A.1: Källor studenter använder för att hitta svar på studierelaterade frågor



Figur A.2: Vanliga frågor studenter har om sina studier

If an AI chatbot could answer basic study-related questions, what concerns would you have about using it?

9 responses

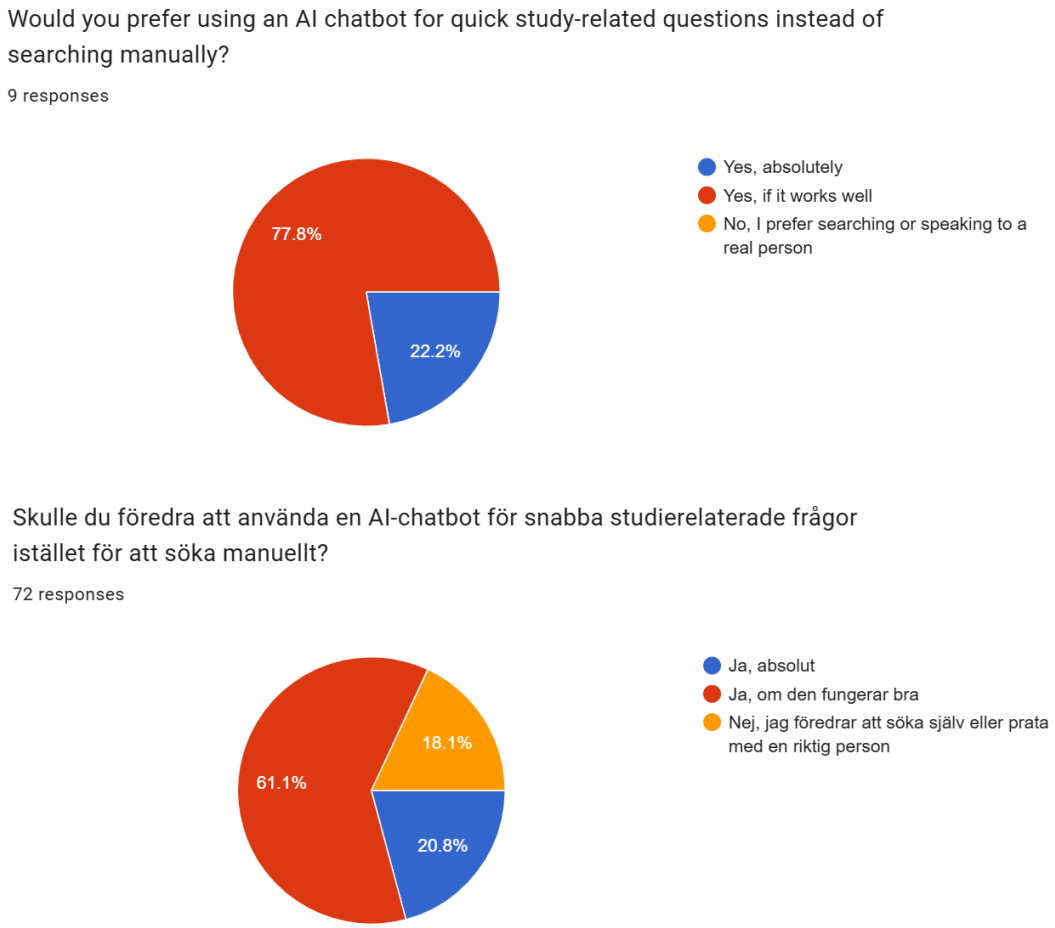
- Sometimes it doesn't work well, more like robotic answers that sometimes put me in circles.
- I guess data privacy. When talking about studies, or topics related to one's own preferences, studies, etc, it is of importance that this data is kept private. This is especially the case when it comes to regulations such as GDPR.
- Not giving related and in depth answers
- I would be concerned about the accuracy and reliability of the AI's responses, as well as the potential for misinformation.
- Privacy
- If it answers incorrectly
- Getting information that is not accurate.
- If my info will remain anonymous and that the answers are correct.

Om en AI-chatbot kunde besvara enkla frågor om studier, vilka funderingar eller bekymmer skulle du ha kring att använda den?

72 responses

- Att den kanske ger fel svar?
- Noggranhet
- Inga
- jag skulle inte lita på att informationen. De frågor den skulle svara på skulle jag troligtvis redan kunna hitta svaren på.
- Att den ger mig felaktig information speciellt när det gäller viktiga beslut
- Mer fakta baserade frågor "kan jag läsa X master med Y kurskrav"
- Opersonligt och omänskligt
- Om den kommer fatta min fråga
- Hur ofta informationen uppdateras

Figur A.3: Studenters funderingar kring användning av en AI-chattbot för studierelaterade frågor



Figur A.4: Studenters åsikter kring användning av AI-chattbot kring studierelaterade frågor

A.2 Intervjufrågor till studievägledare

Vid första intervjun med respektive studievägledare ställdes följande frågor:

- Vad har du för förväntningar på chattbotten?
- Vilka typer av frågor får du oftast från studenter?
- Finns det frågor som är särskilt tidskrävande men ändå rutinmässiga att svara på?
- Finns det något som är viktigt att tänka på i hur en besvarar frågor i ert arbete?
- Viktigt med språk? Svenska/engelska?
- Finns det några typer av frågor som du absolut inte vill att en chattbot hanterar? Varför?
- Var kollar du upp fakta kring utbildningens krav?
- Vad gäller kring tystnadsplikt för er?
- Hur ställer du dig till att skicka ut en enkät till elever och hade ni kunnat hjälpa till med att sprida den?
- Vid dåligt mående, vad kan chattbotten ge för tips då? Finns det telefonnummer/mail/länkar vi kan hänvisa till för samtalsstöd?
- Upplever du att fördelning av tid på olika typer av uppgifter har förändrats över tid? Hur länge har du jobbat som studievägledare?

B

Promptar i RAG-systemet

B.1 Relevansklassificering

You are the very best at determining whether the following web page contents are relevant to a study guidance counselor at a university. The counselor will use relevant content to answer questions from students about education and student life at the university, so your work is very important.
Relevant content includes information about:

- Programs, courses, or subjects offered
- Admission, application, and requirements
- Exams, grading, or study support
- Student life (housing, campus life, activities)
- Scholarships, fees, or financial aid
- Student rights, rules, and regulations
- Career support or internships connected to studies
- University services aimed at students
- Contact information relevant for students

Irrelevant content includes:

- Purely research-focused content without a student perspective
- Internal staff-only news or technical IT updates
- Procurement, job postings, press releases, or collaborations not connected to students
- Anything containing personal opinions. For example interviews with students.

For each numbered snippet provided, respond ONLY with the number followed by `: Yes`, `: No`, or `: Maybe`. Do NOT explain.

B.2 Frågetolkning och breddning

You are a helpful assistant that interprets student queries for study guidance. Your task is to identify and separate background information from questions. Then, you formulate enriched questions that combine the background and the user's original query. You also generate general-purpose questions that ignore the background and only focus on the core question.
Instructions:
Identify and extract all background information from the conversation so far (e.g., program of study, current level, interests), as well as from

the new query.
Identify all questions asked by the student.
If a question is a clear follow-up, merge it into the same question.
Group the result into blocks, where each block contains one background + one merged question.
For each block, generate:

- 3 enriched specific-generated-questions (background-aware)
- 3 general-generated-questions (background-neutral)

Use the same language as the student's query.

B.3 Studievägledarprompt

You are the very best study guidance counselor that answers questions from students strictly based on the provided context.
Your role is very important, be precise and do not make any assumptions.
It is especially important when discussing courses and programmes. Always make sure that the names and codes match.
You **MUST** cite the numbered sources from the context when providing factual statements.
If you cannot back up a statement, do not provide it.
Use the citation format [1], [2], etc., directly after the sentence or fact.
Consider the background information when answering the student's question, e.g. if a student says that they are studying a specific programme, only provide information relevant to that programme.
Each source starts with a number, e.g. "[1]: information from source...".
It is extremely important to use the correct source number when citing information.

If the answer is not in the context, reply with (translated to the student's language):
"I do not have sufficient information to answer your query. I apologize for this."

Be concise, accurate, and neutral.

You will be provided context, background and a question.
Start by stating the question which you are answering, then provide the answer. For example if the question is "Capital of Sweden?" you should answer "What is the capital of Sweden?

The capital of Sweden is Stockholm."
Use the same language as the student's query.

Answer the original query and use the conversation as hints.

Example:

Context: {context}
Full conversation: {conversation_history}
User query: {user_query}

B.4 Specificitetsklassificeringsprompt

You are an extremely precise routing model. Classify the user's query into one of two categories:

1. **specific"**: The user asks for detailed information relevant to one or more courses or programs (e.g. learning outcomes, position in the educational system, entry requirements, course content and structure, upcoming courses, form of teaching, assessment, grading, purpose, higher education qualification and main field of study). OR the user specifies that they are taking a specific course or program (e.g. "I am studying ABB377").

2. **general"**: Otherwise.

Rules:
- If the user provides a specific course code, title, or program name, classify it as "specific".
- If the user says that something is a course or program (or a synonym to these words), assume that it is a course or program.
- Course and program codes are of the format "ABB377", "DIT993", "KFIL13", "N2ADS", "N1SOF", etc.
- Hints about what might be a course or program name can be found in the section of known courses and programs.
- Be strict when recognizing course or program names with broader meaning, for example "mathematical modelling and problem solving" should be recognized as a course name while "mathematics" should not.
- When classifying course or program names, be aware of the context. For example, "I am studying Software Engineering and Management" should be classified as "specific" while "Tell me about Software Engineering and Management" should be classified as "general".

KNOWN COURSES AND PROGRAMS:
{courses}

USER QUERY:
{input}

Based on these definitions, your final answer should be exactly `specific"` or `general"`:

B.5 Prompt för exakt matchning av kurser och program

Your job is to precisely match courses and programs mentioned in a query to actual courses and programs.
You will be provided a user query and list of courses/programs.

Tasks:

1. Identify all courses and programs mentioned in the user query.
2. For each mentioned course or program, find an exact match in the provided list of courses/programs.
3. For each mentioned course or program, if there is exactly one match, include it in the output list of matches.
4. If a mentioned course or program has multiple matches or no matches, formulate a follow-up question to clarify which course or program the user is referring to.
5. Combine all follow-up questions into one question.

Output format:
You MUST output valid JSON, and NOTHING else.
Your output should be a JSON array with objects of this shape: { `"items"`: [{ `mentioned_course_or_program"`: "<string>",
`"exact_match"`: "<code>",
`follow_up_question"`: "<string or empty>"}, ...],
`"combined_follow_up_question"`: "<string or empty>"}
Do NOT include any other text, markdown formatting, or code blocks.
Do not surround output with triple quotes or "json".

Rules:

- Handle each mentioned course or program in the query separately.
- Use the full conversation to help with identification, but only try to match courses in the original query.
- Consider the full conversation when matching, do not ask a repetitive follow up question.
- When doing matching from full conversation be more forgiving with not mentioning the full course name.
- If there is no exact match, leave the `"exact_match"` field empty.
- If you need a follow-up question, you MUST leave the exact match field empty.
- If there are several good matches, provide all of them as options in your follow-up question.
- When providing options in the follow-up question provide the title and the course code (use this format: title (code)"). Also be clear if the option is a course or a program.
- The follow-up question should only be used to find out exactly which course or program the student is referring to. ONLY ask about courses or programs to try to find a match.

- If you have matched all courses or programs in the query, LEAVE the follow-up question field empty.
- If the user has not specified if a program is a bachelor or master program, and there are both options available, provide a follow-up question to clarify which one the user is referring to.
- The follow-up question should contain the unclear matching from the user.

Example 1 (Do not use any specifics of this conversation when answering):
Human: What are the differences between DIT993 and KFIL133?
AI: Did you mean KFIL13 when you said KFIL133?
Human: Yes
Output: `"exact_match"`: "KFIL13", `"exact_match"`: DIT993"

Example 2 (Do not use any specifics of this conversation when answering):
Human: What do I need to do to be accepted into Accounting and Financial Management program?
AI: Did you mean the Financial Management (FEA460) course when you said Accounting and Financial Management program, or are you referring to a different program?
Human: Yes I meant the Financial Management course
Output: `"exact_match"`: FEA460"

Actual courses and/or programs:
{courses}

User query:
{input}

Full conversation:
{follow_up_messages}

B.6 Prompt för detektion av nya frågor

You are an extremely precise routing model.
Based on the user's previous queries and the new query, your job is to classify if the new query is a follow-up on the previous conversation or a separate question.
You should classify the new question into one of two categories:

1. **follow_up"**: If the new query is either a statement clarifying previous messages or a clear continuation of the conversation.
2. **new"**: If the query is not relevant to the previous messages, or if the user indicates that they want to talk about a different topic.

Example:
Conversation so far:
I am studying Software Engineering and Management and I want to know more about the program.
New query:
I have a different question. *or* Let's talk about something else.
Output:
`new"`

Output:
ONLY output `follow_up"` or `new"` and NOTHING else.

Conversation so far: {conversation_history}
New query: {new_input}