

Slutrapport

scAIentific

- *AI-Powered Research Assistant*

Authors: Shirin Meirkhan &
Kalle Andreasson

Semester: VT 2024

1 Sammanfattning

I detta projekt har vi Shirin Meirkhan och Kalle Andreasson, utvecklat en AI-driven forskningsassistent, scAIentific, för att stödja forskare och studenter vid Linnéuniversitetet. Projektets mål var att effektivisera processen för att sammanfatta och identifiera forskningsfronter genom att använda en AI-assistent som kan läsa och analysera upp till 250 vetenskapliga artiklar.

Vi har byggt en teknisk infrastruktur som omfattar användning av C# för backend, Blazor för frontend, och MongoDB för datalagring. För analys och sammanfattning av de vetenskapliga artiklarna använde vi en API-tjänst från OpenAI.

Projektet genomfördes utan formell projektledning, med ett platt arbetsflöde där vi båda bidrog till olika delar av projektet baserat på våra individuella intressen, lärandmål och kunskaper. Projektet levererades framgångsrikt och driftsattes på en cs cloud machine.

Vår insats har mottagits mycket väl av vår kund, Gergei Farkas, en prefekt och lektor vid Linnéuniversitetet. Genom att erbjuda en effektiv och tekniskt avancerad lösning för att underlätta forskningssammanfattningar, har vi kunnat förbättra forskningsprocessens effektivitet.

Contents

| | |
|---|----------|
| 1 Sammanfattning | 2 |
| Contents | 3 |
| 2 Inledning/bakgrund | 3 |
| 3 Syfte och mål | 4 |
| 4 Projektorganisation | 4 |
| 5 Genomförande: metodik, teknik | 4 |
| 6 Resultatbeskrivning/måluppfyllelse | 4 |
| 7 Tekniker som scAIntific använder | 5 |
| OpenAI API:er | 5 |
| Gränssnitt | 5 |
| Backend API | 5 |
| Datalagring | 5 |
| 9 Slutsats – varför blev det så här? | 6 |
| 10 Förslag på vidareutveckling | 6 |
| 11 Förslag till förbättringar inför kommande projekt | 6 |
| Vad gjorde vi bra respektive vad gjorde vi mindre bra? | 6 |

2 Inledning/bakgrund

Hur kan man använda AI i en situation där det är vanligt att forskare och studenter behöver stöd? Detta är särskilt relevant vid skrivandet av uppsatser eller vetenskapliga texter. När forskning om tidigare arbeten ska sammanfattas, är det viktigt att förstå var forskningsfronten befinner sig i förhållande till ämnet. Forskare behöver kunna snabbt överblicka stora mängder tidigare forskning.

Ett exempelprojekt utforskade hur det skiljer sig mellan könen i sättet att knyta kontakter under studieåren. Hypotesen var att manliga studenter kanske knyter mer användbara kontakter under sina högskoleår än kvinnliga studenter. En stor del av detta projekt involverade analys av en omfattande mängd tidigare forskning, baserat på utgångspunkten av 250 vetenskapliga artiklar.

Forskningen identifierades manuellt för att förstå läget inom området. De hade personal som gick igenom abstrakten av dessa 250 forskningsartiklar, försökte identifiera forskningsläget inom området och investerade enorma resurser och tid för att inte bara välja ut artiklarna, utan också läsa, kategorisera dem och skriva om vad forskningen säger om det fenomen de utforskar.

Vi har byggt scAIntific, en AI-driven forskningsassistent som kan läsa och analysera upp till 250 vetenskapliga artiklar och snabbt ge användaren en överblick över forskningsfronten inom ett givet ämnesområde.

3 Syfte och mål

Syftet med projektet var att undersöka hur AI kan användas för att underlätta forskningsprocesser och effektivisera sammanställningen av forskningsfronten. Målet var att utveckla en prototyp som kan analysera och sammanfatta upp till 250 vetenskapliga artiklar eller upp till 512 MB data, och presentera resultaten för användaren via en webbaserad frontend.

Specifika resultat eller prestationer som projektet siktar på att uppnå inkluderar:

1. En prototyp/proof-of-concept för en applikation som använder AI för att sammanställa forskningsfronten.
2. En användarvänlig frontend för att ladda upp artiklar och interagera med AI-assistenten.
3. En robust backend byggd med C# och .NET.
4. En effektiv datalagringslösning med MongoDB.
5. Användning av OpenAI-assistent som använder ChatGPT-4 som modell.

4 Projektorganisation

Uppdragsgivaren för projektet är Linnéuniversitetet, representerat av Gergei Farkas, en prefekt och lektor. Teamet bakom projektet består av två studenter från Linnéuniversitetet: Shirin Meirkhan och Kalle Andreasson. Vi valde att arbeta utan en hierarkisk struktur, vilket vi bibehöll under hela projektets gång. Detta tillvägagångssätt gjorde det möjligt för varje medlem att utnyttja sina specifika färdigheter och kunskaper på bästa sätt.

5 Genomförande: metodik, teknik

Trots att vår projektgrupp slutligen endast bestod av två medlemmar, jämfört med de ursprungligen planerade 5-6, har våra projektmöten och styrgruppsmöten fungerat mycket väl. Mötena har varit effektiva med en lämplig längd som ibland anpassats efter behov beroende på aktuella diskussionspunkter och förberedelser inför kommande vecka. Tidsplanen hölls överlag, även om det förekom små avvikelser vissa veckor till följd av semestrar, personliga åtaganden och andra kurser som vi studerade parallellt med detta projekt. Andra veckor arbetade vi mer än vad som initialt hade planerats. Tidsuppskattningarna var i början av projektet något osäkra, men blev allt mer precisa ju längre projektet fortskred. Vi har inte haft några varningar eller klagomål på varandra. Våra möten har varit frekventa och vi har alltid haft en öppen dialog om allt. Kommunikationen har varit enkel, eftersom vi endast varit två i gruppen, vilket har medfört ett ökat ansvar för oss båda. Vi har genomfört omfattande testning för både frontend och backend. För backend har vi implementerat tester för varje endpoint med olika scenarier, och för frontend har vi genomfört manuella tester. Vi litar på att systemet fungerar.

6 Resultatbeskrivning/måluppfyllelse

Kunden gav oss i uppdrag att undersöka om artificiell intelligens (AI) har förmågan att analysera abstrakt från vetenskapliga artiklar, identifiera samband mellan dessa samt sammanfatta och dra slutsatser från dem. För att möta detta behov utvecklade vi en webbplattform där användare kan ladda upp en fil och skapa en anpassad chattbot för att diskutera innehållet i filen. Kunden ställde inga omfattande krav på webbplatsens funktioner men önskade att designen skulle harmoniera med Linnéuniversitetets övriga webbnärvaro.

Webbplatsen är nu fullt fungerande och erbjuder användarna möjligheten att registrera ett konto, logga in, ladda upp filer och interagera med en chattbot specifikt om det uppladdade dokumentets innehåll. Vi kan konstatera att den artificiella intelligensen klarar av att analysera vetenskapliga abstrakt minst lika effektivt som en manuell genomgång utförd av en person. Dessutom utför AI-analysen på bara några sekunder, jämfört med den avsevärt längre tid det skulle ta för en person att genomföra en likvärdig analys. Enligt vår bedömning har projektets mål därmed uppnåtts.

7 Tekniker som scAIentific använder

OpenAI API:er

scAIentific använder OpenAI:s Assistant API som ger möjligheter att bygga AI-assistenter inom egna applikationer. En assistent har instruktioner och kan använda modeller, verktyg och filer för att svara på användarfrågor. I vår applikation används modellen "gpt-4" för att generera svar till användarens frågor.

Gränssnitt

En klient har utvecklats i Blazor för att tillåta användaren att konversera med scAIentific och ladda upp artiklar.

Backend API

Ett API har utvecklats i C# och .NET för att samla alla systemets olika tjänster bakom en gemensam backend, för att underlätta användningen av dessa tjänster.

Datalagring

MongoDB används för att lagra användarens konton och assistentens egenskaper.

8 Avvikelser/efterkalkyl

Vi har experimenterat och testat assistenten genom vår kod, vilket har skapat problem för andra utvecklare i organisationen. Till exempel tog vi bort alla skapade assistenter i hela organisationen när vi trodde att vi endast raderade vår egen. Vi tog även bort kopplingen mellan assistenten och filen när vi trodde att vi tog bort filen från hela API-minnet. Det skulle ha varit bättre om vi hade fått tillgång till OpenAIs utvecklingsmiljö och kunnat experimentera där, så att vi kunde se vad vår kod faktiskt gjorde. Detta skulle ha sparat tid och minskat problemen.

Vi väntade länge på att IT-avdelningen skulle fixa en driftningsmiljö, vilket hindrade oss från att leverera till kunden i tid. Det skulle ha varit bättre om vi hade driftsatt applikationen själva på en CS cloud machine och levererat tidigt till kunden. Då skulle kunden ha kunnat testa och interagera med assistenten samt hjälpa oss med prompt engineering, för att få reda på vad vi skulle behöva ändra i prompten för att tunna och förbättra assistentens prestanda.

9 Slutsats – varför blev det så här?

Projektet med att utveckla scAIdentific, en AI-driven forskningsassistent, har visat sig vara en framgång trots vissa utmaningar och avvikelser från den ursprungliga planen. Vi kunde uppnå de huvudsakliga målen tack vare en stark samarbetskultur, effektiv kommunikation och en flexibel arbetsmetod. Här följer en sammanfattning av de viktigaste faktorerna som bidrog till vårt resultat:

Positiva Aspekter:

1. **Effektiva Möten och Kommunikation:** Genom att hålla regelbundna och fokuserade möten kunde vi snabbt adressera problem och hålla projektet på rätt spår. Den öppna dialogen inom teamet gjorde att vi kunde dela kunskap och lösa problem tillsammans.
2. **Användning av AI-teknik:** Att integrera OpenAI API för att analysera och sammanfatta vetenskapliga artiklar visade sig vara mycket effektivt. AI-assistenten kunde snabbt bearbeta stora mängder data, vilket sparade betydande tid och resurser.

3. **Robust Teknisk Infrastruktur:** Genom att använda C# och .NET för backend, Blazor för frontend, och MongoDB för datalagring, byggde vi en solid och pålitlig plattform. Detta bidrog till en smidig användarupplevelse och stabil systemprestanda.

Utmaningar och Lärdomar:

1. **Begränsad Tillgång till Utvecklingsmiljö:** Vi stötte på problem när vi testade assistenten i vår produktionsmiljö, vilket ledde till oväntade raderingar av data. Med tillgång till en dedikerad utvecklingsmiljö från början hade dessa problem kunnat undvikas.
2. **Fördröjd Driftsättning:** Vår beroende av IT-avdelningen för att skapa en driftsmiljö försenade projektleveransen. En tidigare och mer kontinuerlig driftsättning hade kunnat ge oss värdefull feedback från kunden och möjliggjort snabbare justeringar och förbättringar.
3. **Anpassning av Assistantens Prompt:** Den sena möjligheten att arbeta med prompt engineering försvårade finjusteringen av assistentens prestanda. Tidigare och kontinuerliga tester med kundens input hade kunnat leda till bättre resultat.

Slutsats:

Trots vissa hinder har vi lyckats skapa en fungerande och effektiv AI-assistent som uppfyller de flesta av våra mål. Denna framgång kan till stor del tillskrivas vårt platta arbetsflöde, starka samarbete och en flexibel projektstruktur. För framtida projekt kan vi dra lärdom av våra erfarenheter genom att säkra bättre utvecklingsmiljöer, genomföra kontinuerlig driftsättning och mer frekvent kundinteraktion.

10 Förslag på vidareutveckling

För att vidareutveckla scAIentific och göra det ännu mer användbart och anpassningsbart för forskare och studenter, finns det flera förbättringsområden som kan utforskas. För det första, skulle möjligheten att ladda upp och analysera Excel-filer vara ett värdefullt tillägg. Detta skulle göra det möjligt för användarna att arbeta med strukturerad data i kalkylblad, vilket ofta används inom forskningsprojekt.

En annan förbättring är att ge användarna möjlighet att skapa och hantera flera assistenter. Detta skulle göra det möjligt för användare att ha specialiserade assistenter för olika forskningsområden eller projekt, vilket kan förbättra effektiviteten och organisationen av deras arbete.

Det är också viktigt att kontinuerligt förbättra och modifiera assistentens prompt för att hålla prestandan hög och anpassad till kundens behov. Detta säkerställer att man får det bästa ur AI-assistenten, särskilt när OpenAI gör uppdateringar och förbättringar i assistentens API.

Slutligen, att ge användarna fler alternativ för att tuna och anpassa assistentens personlighet, egenskaper och färdigheter skulle kunna förbättra användarupplevelsen betydligt. Genom att tillåta sådana anpassningar kan scAIentific bättre matcha användarens specifika behov och preferenser, vilket skulle göra interaktionen mer intuitiv och effektiv.

11 Förslag till förbättringar inför kommande projekt

Vad gjorde vi bra respektive vad gjorde vi mindre bra?

Bra:

- Vi har haft täta möten inom teamet och höll varandra uppdaterade. Vi hjälpte varandra direkt när någon stötte på problem.
- Vi fördelade uppgifterna så att varje person fick en komplett uppgift oberoende av den andra, vilket förhindrade förseningar. Om någon var sen med att utföra sin uppgift, påverkade det inte den andres arbete. Om någon blev klar tidigt, kunde man börja med nästa uppgift.
- Vi testade i god tid, hittade buggar och fixade dem.

Mindre bra:

- Vi driftsatte inte tidigt nog och hade inte kontinuerlig driftsättning, vilket hindrade oss från att leverera tidigt till kunden för att få kontinuerlig feedback. Detta påverkade vår förmåga att bättre förstå kundens behov och att kunna modifiera prompten som skapar assistenten, vilket hade kunnat förbättra assistentens prestanda.