LLM-driven produktivitet - Projektbeskrivning

Allan Gamal

Syfte

Det primära syftet med detta projekt är att utveckla en LLM-baserad personlig AI-assistent. Assistenten syftar till att förbättra produktiviteten och användarupplevelsen genom att erbjuda stöd i dagliga uppgifter, såsom hantering av e-post, schemaläggning av möten, och effektiv filhantering. Genom att använda LLMs för naturlig språkförståelse och -generering, kommer assistenten att kunna förstå och eventuellt eventuellt utföra användarbegäran i en naturlig konversation. Ett annat sätt att se det är att man har en konversation med dina dokument, där du som användare kan ställa frågor om innehållet olika dokument.

Mål

Utveckling av naturlig språkförståelse:

- Implementera LLM för att möjliggöra förståelse av naturligt språk, vilket låter användaren interagera med assistenten via text eller tal.

E-post- och möteshantering:

- Möjliggöra för assistenten att hantera och organisera e-post och eventuellt schemalägga möten skicka e-post.

Granskning och sökfunktionalitet av Användarens Anteckningar och Filer:

- Utveckla assistentens förmåga att granska/söka, organisera och ge insikter om användarens anteckningar och filer, inklusive sammanfattningar baserat på innehållet.
-

Plattformsintegrering:

- Integrera assistenten i en intuitiv applikation. Kanske till och med göra egen hårdvara.

Förväntade Resultat

- En förbättring i hur användare interagerar med sina enheter, vilket ökar deras produktivitet och effektivitet.
- En mer naturlig och mänsklig datorupplevelse som minimerar tid och ansträngning som krävs för att granska digitala uppgifter.

Integrering med E-post och Möteshantering och andra teknologier

Projektet fokuserar på att utveckla en AI-chattbot som fungerar som en interaktiv länk mellan användaren och deras dokument, e-post och kalender. För närvarande utforskar jag utmaningarna för assistenten att utföra handlingar som att skicka e-post och schemalägga tider i kalendern. En handlingsbar AI-assistent är nuvarande en sekudär prioritet i detta skede. En av de huvudsakliga teknologiska komponenterna är Retrieval-Augmented Generation (RAG). RAG-tekniken möjliggör för en LLM att hämta och referera information från dokument som den inte är direkt tränad på, vilket ger assistenten förmågan att leverera relevanta och kontextanpassade svar.

Projektets Potential beskrivet i ett scenario:

En leverans är försenad, och Anders, avdelningschefen i en fabrik, måste identifiera orsaker och hitta en lösning. Den relevanta informationen är spridd över hundratals dokument, e-postmeddelanden och anteckningar. Istället för att granska varje dokument promptar med naturligt språk Anders AIn:

“ Hitta relevanta dokument och kommunikation angående leveransen X från de senaste 3 månaderna”

AI-assitenten använder sin sökfunktionalitet för att genomsöka Anders digitala filer och anteckningar, och har inom några minuter en sammanställd lista över relevanta dokument och e-post. Anders tillsammans med AI-assistenten identifierar förseningen kopplat till en specifik event, och kan snabbt åtgärda problemet.

Plattform

Det verkar som att projektet kommer att genomföras av mig ensam, och jag planerar att utveckla på min Mac, eftersom det är den plattform jag har tillgänglig. Möjligheten att utveckla för Windows kommer att övervägas om resurser och tid tillåter.

Urval av LLM

Efter att ha testat flera LLMs har jag två modeller som presterar väl på min M1 macbook pro med 32gb RAM. Dessa modeller är “orca-2” (13b) och “mistral-instruct-v0.2” (7b). Båda dessa LLMs visar bra resultat och är lämpade för datorn med mina specs. Större modeller som 34b+ parametrar börjar tecken på degradering av prestanda av resursproblem (RAM). Min avsikt är dock att hitta den mest effektiva LLM-modellen som levererar tillfredsställande resultat, genom att välja den minsta möjliga modellen som ändå uppnår rimlig prestanda och resultat, för att ta hänsyn till maskiner med otillräcklig hårdvara.

APIer

För att integrera projektet med kalenderfunktioner kommer jag att använda Google Kalender API, då det är en av de mest använda kalendertjänsterna.
För e-postintegration planerar jag att använda Microsoft Graph API för Outlook och Gmail API för Gmail, eftersom dessa är bland de mest använda e-posttjänsterna.

Poängstatus

Jag har för närvarande samlat 115.5 högskolepoäng och väntar på att kursen 'DAT067 Projekt' ska slutföras. Jag förväntar mig att nå gränsen på 120+ högskolepoäng snart och ser ingen anledning till att jag inte skulle klara kursen.

Exempel på relevanta förkunskaper som är behövliga

Objektorienterad programmering (DAT050):

Grundläggande kunskaper i objektorienterad design och programmering

Agile software project management (DAT257):

Förståelse för agila utvecklingsmetoder, som är viktiga för effektiv projektledning.

Programutveckling (LEU483):

Förkunskaper i att utveckla och strukturera större mjukvaruprojekt med komplexitet.

Objektorienterade applikationer (DAT055):

Fortsatta förkunskaper i objektorienterad programmering, specifikt för applikationsutveckling.

Operativsystem (EDA093):

Förståelse för hur operativsystem fungerar, vilket kan vara relevant för integrering av filhantering