BRAIN x KPMG AI Workshop (WIP)

Introduksjon

Kjell Höcht – leder av investeringsfirmaet Shcüpp Lachvt Capital – har som de fleste andre, skjønt at generativ AI har kommet for å bli. Derfør ønsker han å utvide porteføljen sin til å inkludere selskaper som driver med nettopp det. I forbindelse med dette har han gjennomført en grunding markedsundersøkelse og funnet et potensielt hull i markedet for en personalisert AI-avatar. Han har ikke helt klart å finne ut hvilken form en slik avatar skal ta, og dere har alle muligheter til å designe tema som dere vil.

Kjell er en gjerrig kar og trodde han kunne utvikle de personaliserte avatarene på egenhånd ved hjelp av ChatGPT. Han har derimot innsett at AI er et kompetansekrevende fagfelt og ønsker nå å hyre dyktige konsulenter til den videre utviklingen.

For å oppfylle kravene til Kjell Höcht kreves det at avataren får et spesifikt bruksområde, han ønsker at det skal være mulig å snakke til avataren, og at den svarer deg tilbake. Dersom budsjettet tillater det ønsker han også at avataren er animert.

For å løse den siste biten av oppdraget har Kjell leid inn flere ulike team av konsulenter — dere. I vurderingen av hvem som ender med å vinne implementeringen er det et par aspekter han vektlegger. Blandt disse er:

- Navn på avataren, det er viktig at avataren har et godt navn, slik at produktet selger bra
- Mengden implementert funksjonalitet (kan man snakke til den, snakker den tilbake, er den animert)
- Gir avataren fornuftige svar i forhold til valgt tematikk?

Lykke til!

Lag din egen AI Avatar

Oppgavene går ut på utvikle en egen AI Avatar med utgangspunkt i koden implementert i `avatar.py`.

Azure Openai GPT
Her får dere direkte tilgang til GPT-4, og kan brukes på lik måte som ChatGPT, men gjennom kode!

Azure Speech
Azure Speech har mulighet for å konvertere tale til tekst og tekst til tale med 500 forskjellige stemmer og alle språk du kan tenke deg.

Azure Openai DALL-E

`avatar.py` inkluderer funksjonalitet for å generere bilder. Alternativt kan dere benytte Bing Create

Streamlit

Streamlit er et bibliotek for å enkelt lage webapplikasjoner med veldig lite kode. Streamlit fungerer ved å definere komponenter i koden som for eksempel tekst, bilder, knaper, eller en chat. Streamlit sine sider har oversikt over hvilke komponenter man bruke og hvordan de enkelt kan brukes.

Hvordan kjøre koden

Vi anbefaler å benytte VSCode til oppgaveløsningen, men i prinsippet kan dere bruke en hvilken som helst editor og terminal.

1. Ha Python installert og internettilgang.
2. Pakk ut zip-filen til en egen mappe.
3. Åpne mappen i VSCode og åpne en terminal der (`ctrl+ø`).
4. Skriv disse kommandoene etter hverandre (dobbeltsjekk at du står i korrekt mappe):
    - `pip install pipenv`
    - `pipenv install`
    - `pipenv shell`
    - `streamlit run avatar.py`
5. Dermed vil applikasjonen åpnes i nettleseren, og kan redigeres i `avatar.py` -filen.

For å kjøre applikasjonen for bildegenerering må følgende kommando kjøres:

```
streamlit run image_generator.py
```

Feilsøking

Dersom kommandoene over ikke fungerer, kan følgende løse problemet:

```
pip uninstall virtualenv
pip uninstall pipenv
```

, for så å forsøke metoden i punkt 4.

Oppgaver

Applikasjonen

Bruk de eksisterende komponentene i koden, eller lag deres egne, for å sette sammen for eksempel chat, lyd og bilder til en custom AI-avatar.
Nederst i koden under `WEB APPLICATION` ligger applikasjonen med streamlit-elementene. Her er det lagt inn noen placeholdere som dere kan bruke.

Forslag på elementer som kan inkluderes i avataren ligger under `AVATAR ELEMENTS` i koden. Det er helt valgfritt hvilke elementer man bruker, og det er bare å legge til fler, men her er disse som er lagt opp til i koden:

GPT
Bruk GPT-APIet til å generere svar fra input tekst. Denne inputen kan for eksempel komme fra en chat eller fra brukerens stemme.

Her kan dere bruke `gpt()` funksjonen til å generere et svar. `gpt()` tar inn en liste med messages og en prompt som kommer fra brukeren. Det er vanlig å sende inn hele samtaleloggen (lagret i `st.session_state.messages`), slik at gpt-modellen kan svare utifra konteksten av hele samtalen. Hver message i messages er en dictionary med nøklene `role` og `content`. `role` forteller gpt-modellen hvem som har sagt hva. Selve teksten ligger i `content`. De forskjellige rollene er `user`, som er det brukeren har spurt om, `assistant` er hva gpt-modellen har svart. Til slutt så legges prompten til brukeren til i messages, og det blir sendt via APIet. Mer info om hvordan de forskjellige rollene brukes finnes her.

Prompt engineering
I tillegg finnes `system` rollen som er ekstra instruksjoner til gpt-modellen. Disse kan legges ved i `prompt_engineering()` Denne funskjonen tar inn messages, og legger til system messages. Ekstra instrukser kan også legges ved som `user` meldinger, slik som vist i eksempelet i koden. Bruk denne funksjonen til å finjustere prompten til å passe til temaet dere har valgt på avataren deres.

Mer informasjon om Prompt engineering finnes her og her

Tekst til tale
Gi en stemme til avataren deres! Svarene som genereres fra GPT kan vises som tekst eller konverteres til en stemme gjennom Azure sitt Speech-API. Her kan dere benytte `text2speech()` funksjonen. Denne tar inn en tekst-streng, og snakker med definert stemme og språk. Flere språk og stemmer kan legges til i `voice_languages` og `voice_names`. Alle mulige stemmer og språk finnes her.

Det er også mulig å legge til en speaking style eller humør til noen av de engelske stemmene. Dette kan gjøres ved å benytte `custom_speak()` funksjonen. Her defineres teksten på et format som heter SSML, som muligjør for humør og animering.

Tale til tekst
Istedenfor at inputen skal komme fra en chat, så kan inputen komme direkte fra en stemme. Dette muliggjør at brukeren kan snakke med avataren. Får å gjøre dette, så kan `speech2text()` benyttes, som bruker Speech-APIet til å gjenkjenne stemme fra brukeren og sende generert tekst til GPT. Her må man spesifisere input språk i setup, på samme måte som i text2speech.

Avataren
Bruk egne eller genererte bilder til å være den visuelle avataren. Eksempel prompt: "A cartoon avatar of a personal assistant, only the head, on a white background"

Animasjon
Svarene fra Speech-APIet kan brukes til å animere Avataren slik at munnen beveger seg i henhold til det som sies. I kildekoden finnes en alternativ implementasjon. Denne forutsetter at det ligger bilder med de ulike ansiktsuttrykkene i `/images` -folderen. Ved å laste opp `viseme.psd` i PhotoPea og legge inn avataren dere har laget kan dere legge på og eksportere de ulike ansiktsuttrykkene. Det er i alt støtte for 21 ulike ansiktsuttrykk. For å lage transparent bakgrunn på avataren kan RemoveBg benyttes.

I PhotoPea:

- Last opp `viseme.psd` filen (Open from computer)
- Fra menyen: Velg File-Open for å åpne filen med deres egen avatar. Kopier layer (`ctrl-c`)
- I `viseme.psd` , lim inn avataren som et eget layer (`ctrl-v`)
- Fra menyen: Velg Edit-Transform-Scale for å tilpasse størrelsen på avatern til munnene
- For hver munn (hvert lag) må dere eksportere en `.jpg` filen
- Fra menyen: Velg File-Export as-JPG og lagre filen som viseme-id-[tall fra oppgitt i layer] i 500x500.

Hva vi ser etter

Vi vurderer caset basert på kriterier delt inn i ulike kategorier. Hver kategori blir vurdert individuelt og den endelige vurderingen er en sum av enkeltelementene.

Det visuelle og brukervennlighet
Det visuelle aspeketet sentrerer seg rundt hvordan applikasjonen ser ut. Dette handler om layout, bruk av bilder etc. Hvilken følelse får brukeren av siden og Avataren? Passer det til det valgte temaet, og skjønner brukeren hva denne Avataren er spesialisert til? Er den låst til et språk, eller er den flerspråklig?

Prompt Engineering
Sammen med teksten som sendes inn til GPT så kan man finjustere, legge til tilleggsinformasjon og gi føringer til hvordan svaret skal formuleres. Her vektlegges hva som er gjort for å få Avataren til å passe med valgt tema. Vi vektlegger også kvaliteten på responsen som kommer ut.

Presentasjon og demo
Hvor engasjerende er demoen deres? Pass på at dere har noe som fungerer til dere skal presentere!