# GPT Adjektivhistorie - Rapport Oblig 2 ML

Av: Martin Berg Alstad, Nora Kristiansen og Torbjørn Vatnelid

## Introduksjon

Vi snakket litt om hvilken oppgave vi skulle velge, og landet på oppgave 2 siden den er mest relevant til bacheloroppgaven vår, der vi skal benytte oss av generativ AI for å skrive anbud for bedrifter.

Vi så på mulige tilnærminger til oppgaven og bestemte oss for å bruke OpenAI sitt API for å skrive adjektivhistorier. Vi valgte dette som oppgave fordi *skopet* var lavt nok til at vi tenkte vi kom til å få tid til å eksperimentere en del med prompts, i tillegg til at det kunne skape morsomme historier og dermed være gøy å jobbe med.

Vi testet litt med noen prompts i ChatGPT, for å sjekke om den kunne skrive de slik vi ønsket. Etter å ha testet med prompts direkte i ChatGPT og fått lovende resultater testet vi med å skrive en system message som forklarer hvordan den skal svare.

## Bakgrunn

Generativ AI er en type kunstig intelligens som skaper nye ting ut fra eksisterende materiale. Språkmodeller genererer for eksempel tekster som etterligner hvordan mennesker skriver. Disse modellene lærer fra store mengder eksisterende materiale.

Adjektivhistorier er en type aktivitet der man får en historie der alle adjektivene mangler, så skal de som er med på aktiviteten komme med adjektiver som settes inn på stedene de mangler i historien. Resultatet er ofte humoristisk, da adjektivene ikke gir mening i konteksten. Så vidt vi kan se eksisterer ikke en lignende løsning i dag, det som eksisterer er ferdiglagde historier man kan kjøpe, eller man kan eventuelt skrive slike historier selv.

Dette prosjektet er dermed nyskapende i og med at du kan få automatisk genererte adjektivhistorier på noen sekunder, ut fra eget ønske om tema. Tidligere alternativ har vært å skrive en historie selv, eller å betale noen andre for å gjøre det.

## Metodikk

Vi har valgt å bruke OpenAI sitt API, som er det samme som brukes av ChatGPT. Hvor vi har tilgang til flere Large Language Models. GPT-4-Turbo er den som er mest relevant for vår bruk. API-et kan brukes lett i en Node applikasjon, så vi har valgt å bruke Next.js[1] som kan levere både backend og frontend med React.

For å hjelpe oss med å løse oppgaven har vi brukt ChatGPT ganske mye. Vi har brukt den til å eksperimentere med diverse prompts for historie-generering, da det ikke koster penger i motsetning til OpenAI Playground. Dette har latt oss teste med mange forskjellige prompts uten at det bruker credits vi har brukt på OpenAI sitt API. Vi har også forsøkt å få ChatGPT til å generere system messages som kan brukes i applikasjonen, men disse forsøkene ga ikke resultater som var bedre enn det vi lagde selv.

Det er også brukt til kodegenerering, da vi ikke nødvendigvis er så kjent med React. Da er ChatGPT brukt slik at man ber den generere kode for et spesifikt problem, for å få et utgangspunkt for videre koding. Refererer til eksempel på en prompt og svaret fra ChatGPT [2].
Denne samtalen ble brukt som utgangspunkt for AdjectiveInput-komponenten, og så ble komponenten endret for å tilpasses våre behov.

# Design og utvikling

OpenAI krever at vi bruker API-token for å bruke API-et. Token er hemmelig og skal ikke ligge på klient-siden. Derfor ble vi nødt til å lage en liten backend med et enkelt API som igjen kaller OpenAI sitt API. Det kan gjøres veldig enkelt med Next.js.

Når det kommer til brukergrensesnitt, har vi valgt å holde det svært enkelt med kun et par input-bokser og områder der resultatene vises. Det er forsøkt å gjøre grensesnittet brukervennlig ved å informere brukeren om hva de skal skrive i hver boks. Hovedfokuset med applikasjonen har vært å lage gode prompts slik at vi får gode historier, heller enn å lage et vakkert brukergrensesnitt. Derfor har vi ikke hatt bruk for noen ekstra biblioteker og har kun brukt vanlig React med TypeScript i frontend.

# Eksperimentering og resultat

For eksperimentering har vi brukt OpenAI sin innebygde playground, hvor vi kan eksperimentere med ulike *prompts* og valg av for eksempel temperatur for tilfeldighet.

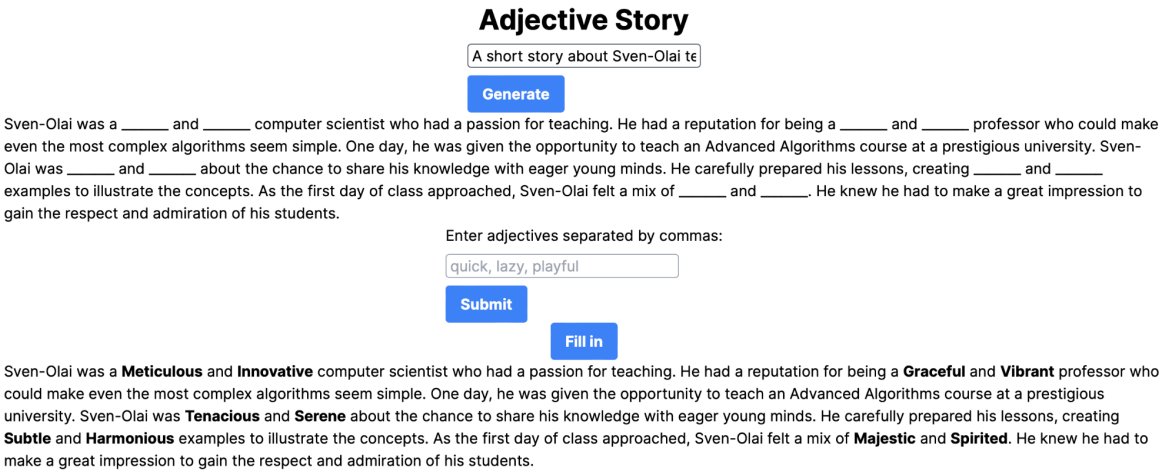
Dette er den første testen med system message [3]:
*You should create an adjective history based on the given prompt.*
*You should just give the answer without any other comments.*
*When creating an adjective history, you should not use any number indicators like (x) next to the missing adjectives.*

Vi brukte stort sett GPT-3.5-Turbo og GPT-4 modellene for testing. Hvor begge kunne gi et riktig svar, men GPT-4 var litt mer nøyaktig. Underveis ble også en preview-versjon av GPT-4-Turbo lansert[4]. Noe som skal gi minst like bra svar som GPT-4, mens den er mye raskere og billigere. Så derfor har vi tatt i bruk den etter den ble lansert.

Vi har også brukt GPT-4 og GPT-4-Turbo for å sette inn adjektiv i historien. Dette var system message som ble brukt:

*You should fill in the missing adjectives denoted by "_____" using the provided adjective list.*
*You should return only the filled out story without any other comments.*
*You should fill the first word in the first missing adjective spot, the second word in the second spot, and so on.*
*You should not try to make the adjectives make sense.*

Her har vi spesifisert hvor og hvordan adjektivene skal settes inn i historien. Hvis disse setningene ikke var med i meldingen forsøkte modellen å sette inn ordene på steder i teksten der det ga mening, som ikke er det vi er ute etter i en adjektivhistorie.




Et eksempel på adjektivhistorie før og etter utfylling, midt i utviklingen.

# Utfordringer og problemløsning

Vi støtte på noen problemer ved bruk av GPT-3.5 i løpet av oppgaven. Ett av problemene er at modellen ikke er så flink til å telle, så historiene ble av og til generert med feil antall adjektiv hvis vi prøvde å sette inn en begrensning på hvor mange adjektiv vi ville ha. Dette er forøvrig et problem med GPT-4-Turbo også, når mengden adjektiver blir stor nok (ca 20 eller fler fant vi ut med eksperimentering). Det virker som at jo flere adjektiver man ber om, jo mindre nøyaktig blir de i svarene sine. Et annet problem er at den ofte bytter ut andre ting enn kun adjektiv, spesielt navn. Den siste utfordringen vi møtte på var at den ikke var konsekvent med måten den byttet ut adjektiv på, selv om vi ga den instrukser på hva som skulle settes inn der adjektivene skulle være. Hvis vi ga instruksjoner om å bytte ut adjektiver med “_” kunne den returnere en historie der adjektiv var byttet ut med “_(adjective)_”.

En annen utfordring var at GPT ikke alltid ville sette inn adjektivene som ble gitt på riktig plass. Av og til satt den inn ord på feil plass eller brukte egne ord dersom den ikke fikk nok som input. Dersom det ikke er nok ord som input, skal den gjenbruke de i samme rekkefølge som de ble gitt i.

GPT-4 modellen gir generelt bedre svar, men av og til så har den også gitt svar som er helt feil. Noe som var mulig å forbedre ved å tilpasse system message. For eksempel i noen tilfeller skrev den (1), (2) osv. etter manglende adjektiv. Noe som kunne fikses ved å legge til en melding i system message, som ber den  om å ikke legge til (x) bak adjektiv.

# Diskusjon

Denne applikasjonen gir svært lite reell verdi. Den gir underholdning og er et morsomt lite prosjekt som kan gi litt latter i hverdagen.

Når det kommer til ytelse ble den svært mye bedre etter at vi byttet fra GPT-4 til GPT-4-Turbo. Det tar oftest et par sekunder å generere/fylle inn historier, resten av applikasjonen reagerer umiddelbart. Før vi byttet til GPT-4-Turbo kunne generering ta opp til ett minutt.

Med tanke på etikk står vi overfor mange av de samme problemene som alle språkmodeller. Særlig er det en risiko for at modellen kan generere støtende historier, eller inneholde feil informasjon. GPT har allerede innebygde filtre på ting som vold, ulovlige aktiviteter, rasisme og seksuelt innhold.

# Reproduserbarhet

I dette prosjektet har vi brukt github som versjonshåndtering, som gir oss mulighet til å kunne dele prosjektet med åpen kildekode. I github har vi brukt branches til å håndtere forskjellige versjoner. Prosjektet startet med at vi alle lagde hver vår versjon, før vi landet på en som vi alle fortsatte å jobbe videre med. Normalt sett ville vi har brukt branching til å løse enkelte oppgaver, som får så bli flettet inn i “master” branchen. Det er blitt forsøkt å gi så konkrete commit beskjeder som mulig, slik at alle skal kunne vite hva som er blitt gjort når. Vi valgte denne måten å jobbe på for at alle på gruppen skulle få prøve seg på å lage en egen utgave. Videre gav denne metoden oss muligheten til å eksperimentere og lære på egen hånd. Når det kommer til dokumentasjon er dette gjort som kommentarer i koden, og i README.

# Konklusjon

Prestasjonen med oppgaven er at vi har tatt i bruk et rammeverk som de fleste på gruppa ikke har brukt før og laget et fungerende produkt med det. Vi valgte denne oppgaven mest fordi den er relevant til bacheloroppgaven, hvor vi skal jobbe med OpenAI sitt API. Vi har fått verdifull erfaring innen bruk av OpenAI sine verktøy som gir oss en fordel når vi starter med oppgaven.

# Framtidig arbeid

Vi ønsker å gjøre applikasjonen litt mer brukervennlig med flere valg, som mulighet til å velge antall adjektiv du vil måtte fylle inn og lengde på historien. Vi ønsker også å lage penere design, med muligheten for å vise teksten underveis, i stedet for å vise alt når all data er mottatt.

Slik det er nå så kan man fylle inn hvilke som helst ord. Det blir ikke verifisert på noe vis at ordene som blir sendt inn er adjektiv. Vi kan oppdatere GPT-modellen for å be den verifisere at ordene er adjektiv. Også eventuelt ignorere de ordene eller be brukeren fylle inn andre ord.

# Referanser


[1] «Next.js by Vercel - The React Framework». Åpnet: 13. november 2023. [Online]. Tilgjengelig på: https://nextjs.org/

[2] «Inputting Adjectives in React», ChatGPT. Åpnet: 13. november 2023. [Online]. Tilgjengelig på: https://chat.openai.com/share/5e65712f-eec6-4a6e-b1a8-c44519ef3690

[3] «Enchanted Forest Fountain Quest», ChatGPT. Åpnet: 13. november 2023. [Online]. Tilgjengelig på: https://chat.openai.com/share/febefa51-06d8-41b5-8390-f2a803f1f16e

[4] «GPT-4 Turbo | OpenAI Help Center». Åpnet: 13. november 2023. [Online]. Tilgjengelig på: https://help.openai.com/en/articles/8555510-gpt-4-turbo