

Uke+07/IS-105ICA04Waves

Gruppe 1, BIOS Brothers:

Anders Eskeland – Haris Saric – Magnus Hornnes –

Sarvan Naganathar – Sebastian Nomme – Sindre Schjølberg

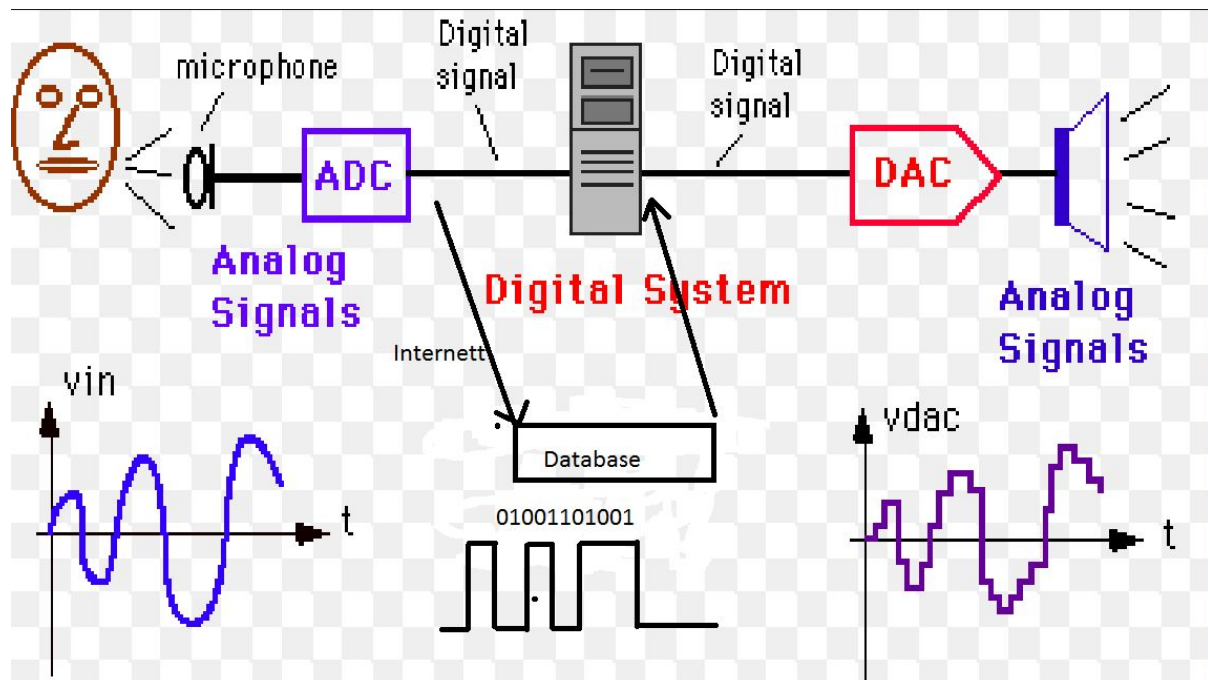
Syntetisk generering og gjenkjenning av naturlig språk

Det aller første som skjer under denne prosessen er at vi mennesker skaper lyd ved å vibrere på stemmebåndene. Lyden vi produserer blir “sendt” ut som analoge signaler som kan bli fanget opp av bl.a. mennesker og mange forskjellige maskiner. I vårt eksempel skal vi få en datamaskin til å fange opp disse signalene, og derfor trenger vi en maskinvare som kan klare det, og vi bruker en helt standard mikrofon fra en bærbar datamaskin. Som sagt sender vi ut signaler når vi snakker, og disse signalene blir sendt ut mot mikrofonen. Når signalene våre går inn i mikrofonen, treffer de en tynn membran som beveger seg frem og tilbake i forhold til de analoge signalene vi har lagd ved bruk av vårt stemmebånd. Membranen tar imot de analoge signalene i form av energi. På membranen i mikrofonen henger det også en spiral som beveger seg på lik måte som membranen. Denne spiralen gjør det mulig å videreføre energien gjennom mikrofonen fra membranen der den ble mottatt. Resultatet blir et audiosignal som er en representasjon av lyd, og vanligvis i form av elektrisk spenning. Denne elektriske spenningen kan bli lagret i forskjellige formater og kan bli brukt på mange forskjellige områder, for eksempel kan den bli brukt som en forsterker. Man tar inn originallyden og øker størrelsen på lyden og sender den ut igjen gjennom en høyttaler.

I vår chatterbotmodell bruker vi en kode i python der vi tar i bruk et sett med allerede ferdig implementerte moduler som for eksempel “Speech recognition” som støtter blant annet “Google Speech Recognition” og “Pyttssx” som støtter vanlig tekst til tale. Dette gir oss muligheten til å direkte bruke *ADC (Analog-to-digital converter)* i Python-koden vår slik at vi slipper å bruke tredjepartsprogrammer. Når det analoge signalet har nådd mikrofonen og blitt bearbeidet til audiosignal, vil signalet til slutt bli gjort om til digitalt signal via “ADC”. Dermed kan dette digitale signalet bli sendt til en annen server, i vårt tilfelle vil den sende data til Google-serverene. Googles software kan analysere vår tale, fordele den i små biter for at audioanalysen kan finne ut av hva som blir sagt. Sofistikerte og avanserte algoritmer analyserer talen for å finne ut om hvilke ord som skal bli tatt ut av databasen, og ord som høres like ut som “meat” og “meet” må bli sammenlignet iforhold til helheten av setningen

for å finne det passende ordet til talen. Etter vi har sendt det til Google sine servere vil de da sende tilbake til oss om ordene vi har sagt ble kjent igjen som ord, eller om den ikke klarte å tolke det vi prøvde å si. Denne koden vil da bli konvertert tilbake via DAC (Digital-to-analog converter) og deretter spiller den ut lyden via høyttalerne (ofte høyere lyd på grunn av økt størrelse av lyden).

Illustrasjon



Konklusjon

Grunnet at kvaliteten på mikrofon på bærbar pc ikke er særlig bra, så har det vært litt problemer med å få jevn tester. Skulle man ha jobbet mye med dette ville nok en ekstern mikrofon vært en anbefaling. Vi har i tillegg oppdaget at riktig tydelig artikulasjon er viktig, dialekter gjorde eksperimenteringen vanskeligere, og man må snakke relativt monotont i forhold til hva man vanligvis ville ha gjort.

Referanser:

<http://www.explainthatstuff.com/microphones.html>

Utlevert eksempelkode på fronter, soundcheck.py og speechrecogn.py

MIT dokumentene fra fronter