Gruppe 04
22.02.16

ICA04 - Waves

Jevn utvikling

Stemmegjenkjenning er ett område som har blitt aktivt forsket på i underkant av 100 år nå. Fra tidlig av ble det utviklet systemer som kunne gjøre om ett og ett ord men som ofte hadde feilstavinger og ellers lite ord i vokabularet. Videre opp gjennom årene ble det utviklet nye systemer og algoritmer for å minske feilene i tilegg til å øke vokabularet. Hovedproblemet til forskerene har lenge vært tekniske begrensinger men i de siste tiårene har datamaskin teknologien hatt en ekstrem utvikling og dermed har det gitt et løft i forskning på dette området.

Lydkortet til PCen konverterer analoge lydbølger som fanges opp av mikrofonen til et digitalt audioformat. Talegjenkjenningsprogrammet prosesserer audio, isolerer lydsegmenter som sannsynligvis er tale, og konverterer dem til numeriske verdier, altså dekoder det den gjenkjenner som å være tale til et språk den kan forstå, digitale bits.

Den akustiske modellen til programmet bryter tale ned i fonemer, hver enkelt lyd som har blitt uttalt. Disse er definert på forhånd, så programmet vet at lyden vi kjenner som “t” hører sammen med bokstaven “t”.

Språkmodellen sammenligner fonemene med ord som er definert i den innebygde ordboken. Den setter i praksis sammen fonemene til en string og avgjør hvilke(t) ord dette sannsynligvis er.

Det finnes ulike metoder for å gjøre dette, den kan prøve å gjenkjenne ord i sin helhet, eller gjenkjenne ord ved hjelp av mønstre hvor den bryter hvert ord ned i bits og ser på forskjellige attributter, som f.eks hvor mange vokaler ordet inneholder. Programmet kan også bruke statistisk analyse og kunnskap om grammatikk til å avgjøre sannsynlighetsgraden for at visse ord følger andre og dukker opp i en viss sammenheng.

Et stort problem med talegjenkjennelse er at mennesker ikke alltid snakker likt, og for å gjenkjenne en lyd er datamaskinen avhengig av å vite hvordan den skal høres ut på forhånd. Når folk snakker i forskjellig tempo eller med ulik dialekt blir det dermed vanskeligere for programmet å fastslå hva som har blitt sagt. Nøyaktig hvilken metode som bør benyttes for å gjenkjenne talen er derfor helt avhengig av hva som er formålet til det individuelle programmet.

Mange programmer er også avhengig av feedback fra brukere. Programmet gjetter hva det tror vi har sagt, og ber om en bekreftelse fra brukeren på at dette er riktig eller feil. Når brukeren korrigerer programmet vil det over tid lære å gjenkjenne talestilen til den spesifikke brukeren, og slik trenes programmet til å bli mer effektivt. Denne metoden brukes f.eks i diktafoner.

De fleste moderne talegjenkjenningssystemer prosesserer audioen per "frame", hvor en frame er ca 10ms.

Det kan også brukes formanter til å identifisere hver enkelt karakter (hver bokstav). Dette er spesielt effektivt på vokaler, selv om de uttales i forskjellige tonehøyde. Resonnsansen til hver enkelt karakter blir fanget opp og karakteren kan bli bestemt basert på amplitude (formant f1, frekvens (formant f2), og forolder mellom disse (f2-f1). Dette gjøres mulig ved at hver karakter har en gjennomsnittlig verdi for hver formant, og ved at ingen karakter har to like formanter; Det er mulig å ha en lik formant, for eksempel på dette er "Æ" og "Ø" som har har lik frekvens, men ikke to.

Formant syntese bruker ikke opptak fra mennesker men heller datagenerert output. Dette gjør at det ofte høres kunstig og syntetisk ut, men har fordeler som at det blir lett forståelig og ikke er avhengig av taleprøver, samtidig som det kun krever en begrenset mengde datakraft. I mer avanserte former kan det brukes til å gjengi følelser og toner i talen. Formant syntese brukes derfor ofte for navigering på PC for synshemmende, og i integrerte systemer.

Siden språket vårt inneholder flere millioner ord, vil det kreve enormt mye av maskinen å søke opp hvert ord i en database med alle sammen. For å minske kravet for datakraft nødvendig for å gjennomføre oppgaven, anvendes det sannsynlighetsregning. Ved å minske informasjonsmengden for forskjellige ord og uttrykk reduserer man kravet for datakraft. Eksempler på dette er for eksempel at det er mer sannsynlig at en bruker vil si: "Universitetet i Agder", enn: "Universitetet i potet." På samme måte kan man sette opp programmet med en viss forståelse for grammatikk, slik at programmet selv kan anta hvilke ord som kommer etter hverandre, og dermed minske datakapasiteten.

Kilder:

Wikipedia. (sist endret 27. Des 2015) *Talegjenkjenning*. Tilgjengelig fra: https://no.wikipedia.org/wiki/Talegjenkjenning

Wikipedia. (sist endret 21.02.2016, klokken 19:58) *Speech recognition*. Tilgjengelig fra: https://en.wikipedia.org/wiki/Speech_recognition

Mediacollege. (ukjent årstall) *How Microphones Work.* Tilgjengelig fra: http://www.mediacollege.com/audio/microphones/how-microphones-work.html

Wikipedia. (sist endret 08.01.2016) *Mikrofon*. Tilgjengelig fra: https://no.wikipedia.org/wiki/Mikrofon

Microsoft. (ukjent årstall). *how speech recognition works (Microsoft Speech)*. hentet 21.02.2016, fra https://msdn.microsoft.com/en-us/library/hh378337%28v=office.14%29.aspx

Sudeesh Puthiyedath (2006). *how speech recognition works*. hentet 21.02 2016, fra http://www.codeguru.com/cpp/g-m/multimedia/audio/article.php/c12363/How-Speech-Recognition-Works.htm

Chris Woodford. (2015). *voice recognition software.* Hentet 21.02 2016, fra http://www.explainthatstuff.com/voicerecognition.html

Wikipedia. (29. october 2015 kl 22:25). *Formant*. Hentet 21.02 2016, fra https://en.wikipedia.org/wiki/Formant

Wikipedia. (16. februar 2016 kl 0807). *Speech synthesis.* Hentet 1.02 2016, fra https://en.wikipedia.org/wiki/Speech_synthesis#Formant_synthesis