Verbeteren van VR stottertherapie door het herkennen en verwerken van verbale antwoorden met spraakherkennings software.

Jona De Neve.

Scriptie voorgedragen tot het bekomen van de graad van
Professionele bachelor in de toegepaste informatica

Promotor: Mevr. Lena De Mol
Co-promotor: Mevr. Jana Van Damme
Academiejaar: 2022–2023
Tweede examenperiode

Departement IT en Digitale Innovatie .

HO
GENT

Woord vooraf

Deze bachelorproef werd geschreven in het kader van het voltooien van de opleiding Toegepaste Informatica afstudeerrichting Mobile & Enterprise development. Hoewel dit onderwerp niet valt onder mijn afstudeerrichting, koos ik ervoor uit interesse van de huidige stand van Artificiële Intelligentie en hoe het verder evolueert in de toekomst. Zo volg ik de vooruitgang ook wekelijks. Daarnaast wil ik graag enkele mensen bedanken, zonder wie deze proef niet tot stand zou zijn gekomen.
In de eerste plaats wil ik mijn promotor, Lena De Mol, bedanken voor de begeleiding en feedback doorheen mijn bachelorproef.
Daarnaast wil ik mijn co-promotor, Jana Van Damme, bedanken voor de hulp tijdens de opzet van deze bachelorproef en voor alle nuttige feedback onderweg.
Ook wil ik mijn ouders bedanken omdat ze mij bleven aansporen om door te zetten.
Als laatste wil ik mijn familie en vrienden bedanken voor de hulp die ze aanboden.

Samenvatting

Het 360° Zorglab van de Hogeschool Gent biedt stotterpatiënten, aan de hand van interactieve video's met Virtual Reality, kansen aan om in realistische situaties met stotteren om te leren gaan. Per scenario bestaat er een filmpje dat in fragmenten is opgedeeld maar om naar het volgende fragment te gaan moet de gebruiker zelf zijn antwoord nog manueel aanduiden. Dit haalt de dynamiek uit de oefening weg. Daarom wordt er in deze bachelorproef informatica gezocht naar een oplossing om dit te verhelpen. Om dit te verwerven wordt eerst een literatuurstudie uitgevoerd. Deze bevat informatie over wat het stotteren is en veroorzaakt, de huidige toestand en geschiedenis van Virtual Reality en de stand van Artificiële Intelligentie met een uitgebreidere analyse van spraakherkenning en natuurlijke taalverwerking. Als tweede wordt er lijst opgesteld van bestaande spraakherkennings- en natuurlijke taalverwerkingssoftware. Vervolgens wordt er een proof of concept opgesteld en aan de hand daarvan verschillende spraakherkenningsmodellen vergeleken op spraakfragmenten met en zonder stottermomenten. Uit de resultaten valt op dat Whisper's transcripties de verstaanbaarheid van gesproken fragmenten behouden in zowel het kleine als medium model, ondanks enkele herkenningsfouten. Het kleine model verwerkt 3 à 4 keer sneller, maar met meer fouten. Google ASR is over het algemeen trager dan Whisper, met nauwkeurigheidsinvloeden van stottermomenten en gebrek aan leestekens. Microsoft Azure speech services zijn sneller dan Google, maar minder nauwkeurig bij stotterende spraak door herhaalde woorden. Over het algemeen blijkt het medium Whisper-model het meest geschikt, ondanks langere verwerkingstijd dan Azure en kleine Whisper-model, vanwege consistente nauwkeurigheid. De betaalde services als Google en Microsoft presteerden minder goed dan verwacht bij stotterfragmenten. Het onderzoek biedt waarde voor het 360° Zorglab en kan dienen als basis voor verdere toepassingen en onderzoek naar spraakherkenning bij stotteren.

Inhoudsopgave

Lijst van figuren

1

Inleiding

In de reclamespot 'The Impact Will Be Real' over de Metaverse toont Meta (2022) hun visie over de rol die Virtual Reality (VR) speelt in de toekomst. Zo laten ze verschillende toepassingen ervan in het onderwijs zien. Jammer genoeg staat onze technologie nog niet zo ver als wat er in de reclame gezien kan worden maar ook nu al vindt VR zich een baan in verschillende opleidingen.
De Hogeschool van Gent maakt ook gebruik van VR om studenten de kans te geven in meer realistische situaties te oefenen. Hiervoor zijn twee verschillende technieken gebruikt. Allereerst heb je het renderen van een omgeving. Dit laat de gebruiker een interactieve wereld van 3D modellen ontdekken. Zo bestaan er drie virtuele kamers waarin de student kan oefenen. De andere manier is aan de hand van een 360° opname die wordt gemaakt aan de hand van een 360° camera. Omdat dit een opname van de werkelijkheid neemt, ziet deze methode er realistischer uit.
Dit is waar er op het probleem wordt gestoten. Aangezien de tweede methode werkt met een opname moet er naar verschillende fragmenten gesprongen worden naargelang het antwoord dat de gebruiker ingeeft. Dit wordt handmatig gedaan door een begeleider. Hierdoor staat de oefening tijdelijk stil wat de echtheid van de situatie weghaalt. Daarom wordt in deze bachelorproef toegepaste informatica onderzocht hoe het overschakelen anders kan aangepakt worden zodat het voor de gebruiker realistischer aanvoelt. Hiervoor kijken we richting Artificiële Intelligentie (AI).
Het afgelopen jaar is de populariteit van AI enorm gestegen. Van text-to-image models zoals DALL-E 2, Imagen en Stable diffusion die een gegeven tekst prompt kunnen omzetten in afbeeldingen tot voice-models die stemmen kunnen imiteren, zo goed als iedereen heeft al gehoord van AI. Meestal is het jammer genoeg het negatieve aan AI dat de ronde doet zoals het verspreiden van misinformatie aan de hand van deepfakes of het schenden van auteursrechten maar afgezien dat heeft

het veel mogelijk heden om de mensheid te helpen.

1.1. Probleemstelling

Wanneer de stotterpatiënt antwoordt op de gestelde vraag wordt er niet automatisch overgeschakeld naar een volgend fragment. Er moet namelijk handmatig geklikt worden op het gewenste hoofdstuk door iemand die de oefening kent. Dit zorgt dat er steeds een begeleider aanwezig moet zijn. Ook zal de dynamiek van de oefening verbroken worden omdat het pas verder kan gaan wanneer de begeleider het juiste fragment vindt.

1.2. Onderzoeksvraag

Om dit probleem te verhelpen wordt hiervoor gekeken hoe AI te hulp kan schieten om stotterpatiënten een meer interactieve ervaring geven. Om dit te realiseren zullen de volgende vragen beantwoord worden:

- Hoe gaat de applicatie registreren wat er werd gezegd?
- Wat kan er mislopen bij het registreren van spraak?
- Welke aspecten van stotteren hebben effect op de spraakherkenning?
- Hoe kiest de applicatie een volgend fragment op basis van de gegenereerde tekst?

1.3. Onderzoeksdoelstelling

In deze bachelorproef wordt een proof of concept opgesteld om te kijken hoe zo een AI-applicatie te werk zou kunnen gaan. De bedoeling van deze applicatie is eerst en vooral het registreren wanneer de gebruiker aan het antwoorden is. Vervolgens transcribeert het wat er gezegd wordt. Als laatste gaat het op basis van de context een gepast fragment zoeken om verder mee te gaan.

1.4. Opzet van deze bachelorproef

De rest van deze bachelorproef is als volgt opgebouwd:
In Hoofdstuk 2 wordt een overzicht gegeven van de stand van zaken binnen het onderzoeksdomein, op basis van een literatuurstudie.
In Hoofdstuk 3 wordt de methodologie toegelicht en worden de gebruikte onderzoekstechnieken besproken om een antwoord te kunnen formuleren op de onderzoeksvragen.
In Hoofdstuk 4 worden de resultaten besproken.
In Hoofdstuk 5, tenslotte, wordt de conclusie gegeven en een antwoord geformuleerd op de onderzoeksvragen. Daarbij wordt ook een aanzet gegeven voor toekomstig onderzoek binnen dit domein.

2

Stand van zaken

Zoals vermeld in de inleiding heeft het 360°-Zorglab van Hogeschool Gent verschillende doeleinden. Zo kunnen studenten geneeskunde leren hoe ze een operatie moeten uitvoeren zonder iemand als testpersoon te gebruiken. Anderzijds kan het een persoon die plankenkoorts heeft een presentatie of een toespraak leren houden. In deze situatie zal een patiënt met een stotter leren een vlot gesprek te houden wanneer hij bijvoorbeeld op sollicitatie gaat. Dit wordt gerealiseerd aan de hand van 360° opnames die voor de patiënt opnieuw afgespeeld kunnen worden in VR.

2.1. Stotteren

Stotteren is een spraakstoornis die gekenmerkt wordt door onderbrekingen en verstoringen in het vloeiend spreken. De aandoening komt vaker voor bij kinderen en gaat bij 80% over naarmate ze opgroeien (Gordon, 2002). Er zijn verschillende types van onvloeiendheid in spraak (Chee e.a., 2009): tussenwerpsels (woorden zoals 'uh' of 'eh'), herzieningen (het aanpassen van de inhoud of de grammaticale structuur van de zin), onafgemaakte zinnen, herhaling, aanhoudende geluiden en afgebroken woorden. Een stotter kan voorkomen in verschillende mate, sommigen zullen meer last hebben zichzelf te uiten terwijl bij anderen de stoornis niet hard opvalt.
De specifieke oorzaak van een stotter is nog niet doorgrond. Uit het onderzoek van Gordon (2002) blijkt echter dat de rechter- en linkerhersenhelft verschillende en tegenovergestelde rollen lijken te spelen bij het ontstaan van stotteren. Symptomen van stotteren worden geassocieerd met activering van de voorste gebieden in de linkerhersenhelft, terwijl zowel voorste als achterste perisylvische gebieden in de rechterhersenhelft geactiveerd worden wanneer het spreken vloeiender wordt. Deze verbetering kan mogelijk te wijten zijn aan de koppeling van motorische en zintuiglijke gebieden in de rechterhersenhelft.

(a) The Sensorama machine

Source: https://www.historyofinformation.com/detail.php?id=2785

(b) Ivan Sutherland's first VR Head Mounted Display, The Sword of Damocles

Source: https://www.dsource.in/course/virtual-reality-introduction/evolution-vr/sword-damocles-head-mounted-display

2.2. Virtuele realiteit

Hoewel Virtual Reality tegenwoordig ver gevorderd is, bestaat het al voor een lange tijd. Het eerste toestel dat de werkelijkheid nabootste was Morton Heilig's Sensorama[1] (Figuur 2.1a) uit 1962. Dit liet de gebruiker ervaren hoe het voelde om op een motor door Boston te rijden. In de jaren nadien werden ook andere toestellen uitgevonden zoals de 'Sword of Damocles'[2] (Figuur 2.1b) die de locatie van het hoofd en de ogen volgde en Nintendo's 'power glove'[3] voor de NES (Boas, 2012).
De eerste vermelding van de term VR daarentegen kwam pas rond 1985 zegt Bryson (2013). Hij verteld over hoe Jaron Lanier de term 'virtual reality' gebruikte om het Virtual Interactive Environment Workstation (VIEW) lab van NASA te beschrijven. De daaropvolgende jaren werd de term steeds populairder met tot gevolg dat het ruim werd toegepast op verschillende toepassingen. Daarom moest de term VR goed gedefinieerd worden:

> Virtual Reality is the use of computer technology to create the effect of an interactive three-dimensional world in which the objects have a sense of spatial presence. (Bryson, 2013)

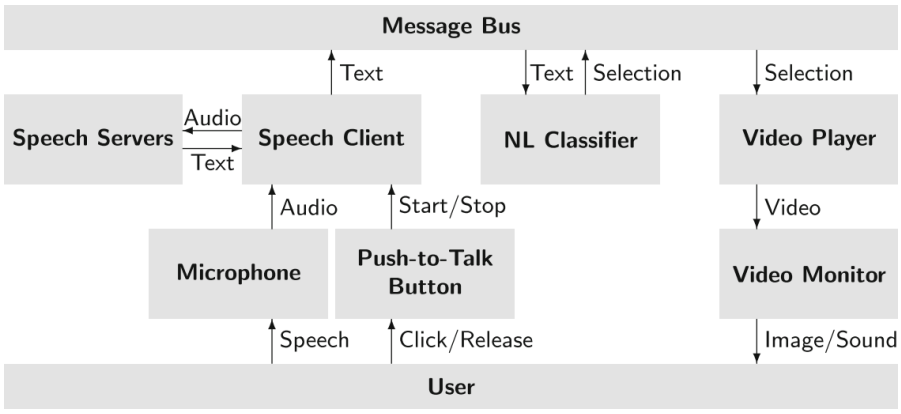
Tegenwoordig zijn VR-toestellen te vinden in vele soorten een maten. Zo zijn er headsets gemaakt voor PC of consoles en anderen voor smartphones. Dankzij software- en hardwareverbeteringen bestaan er nu zelfs headsets die autonoom

[1] https://www.historyofinformation.com/detail.php?id=2785
[2] https://www.dsource.in/course/virtual-reality-introduction/evolution-vr/sword-damocles-head-mounted-display
[3] https://en.wikipedia.org/wiki/Power_Glove

werken. Dit zorgt ervoor dat kabels en complexe set-ups niet nodig zijn. Deze verbeteringen zorgen ook dat de digitale wereld realistischer aanvoelt.

2.3. Artificiële intelligentie

Het afgelopen jaar is de populariteit van AI enorm gestegen, zoals text-to-image models zoals DALL-E 2, Imagen en Stable diffusion die een gegeven tekst prompt kunnen omzetten in afbeeldingen. Daarnaast zijn er ook applicaties die language models gebruiken zoals Chat-GPT, spraak assistenten en Github Copilot. Zij kunnen teksten verwerken en accurate antwoorden genereren op basis van hun kennis ook al zijn deze niet altijd accuraat (Meer hierover in 2.3.2)
Om het overschakelen naar een volgend fragment te laten gebeuren zonder begeleiding en zonder veel tijd te verliezen, zouden verbale antwoorden gegeven kunnen worden. Een mooi voorbeeld van een gelijkaardige interactie is het project 'Dimensions in Testimony' van de USC Shoah Foundation (2020). Daar kunnen bezoekers vragen stellen aan een overlevende van de Holocaust die op voorhand een volledig interview heeft afgelegd. Dit werd mogelijk gemaakt dankzij het systeem dat erachter zit (Figuur 2.2). Dit systeem is opgebouwd uit volgende componenten: een software voor spraakherkenning (ASR) die de gebruikers verbale vraag in tekst omzet, een Natural Language Classifier (NLC) die op basis van de gegenereerde tekst een antwoord via een audio/video fragment voorziet en een mediaspeler die de fragmenten kan afspelen met tussendoor een inactieve animatie (Traum e.a., 2015).



Figuur (2.2)
Dimensions in Testimony - Systeem Architectuur (Traum e.a., 2015)

Een ander voorbeeld zijn de home assistenten zoals Google Home, Amazon Echo of Apple HomePod. Zij staan op stand-by tot de gebruiker het triggerwoord zegt, 'Hey Google' bijvoorbeeld. Eenmaal geactiveerd luisteren ze naar wat de gebruiker zegt en zetten ze dit aan de hand van een ASR om naar tekst. Daarna haalt het met natuurlijke taalverwerking (NLP) de intentie en sleutelwoorden uit de tekst en genereert daarmee een gepast antwoord. Als laatste zet het gegenereerde tekst

om naar spraak.

2.3.1. Spraakherkening

Net als VR is spraakherkenning de laatste jaren veel vooruitgegaan, dit komt omdat er meer data beschikbaar is, de computers sneller en de algoritmes, genaamd deep neural networks, beter zijn (van Hessen, 2020). Deze netwerken worden getraind op grote datasets bestaande uit diverse spraakfragmenten. Hieruit leert het de patronen en kenmerken van menselijke spraak. Maar perfect zullen ze nooit zijn zegt van Hessen (2020). Dit is geen verrassing aangezien mensen zelf nog vaak problemen hebben bij het verstaan van een andere. Er zijn verschillende stappen waar de ASR de fout kan ingaan.

Domeinproblemen

Onder de domein problemen valt ten eerste galm en geluidshinder. Dit komt voor wanneer een opname gemaakt wordt in een omgeving met veel achtergrond lawaai. Net als mensen heeft de ASR moeilijkheden met het verstaan van de spreker wanneer deze overstemd wordt door de omgeving (Alharbi e.a., 2021). Er zijn verschillende manieren om dit probleem te beperken. Zo wordt het aangeraden om het fragment op te nemen in een rustige omgeving. Als het audiobestand toch veel geruis of achtergrondlawaai bevat is het ook mogelijk dit bestand door een stem extractie software te halen zoals Vocal Remover[4]. Het originele doel van deze tool is om de zang van de instrumenten te scheiden in een lied maar dit werkt ook voor gewone spraakbestanden. Naast galm en geluidshinder is het ook moeilijker wanneer twee of meer mensen door elkaar spreken. Dit heet spraak overlapping. Hoe meer mensen in hetzelfde geluidsfragment door elkaar spreken hoe moeilijker het wordt om de spraak te herkennen. Daarom raad van Hessen (2020) aan om een microfoon per spreker aan te brengen. Dit zorgt ervoor dat er voor elke spreker een ander geluidsfragment beschikbaar is om te transcriberen. Een derde domein probleem is domain mismatch. Dit houdt in dat er een discrepantie is tussen het domein gebruikt voor het model te trainen en de use case waar het model in wordt gebruikt. Zo zijn bijvoorbeeld nieuwe accenten of andere omgevingen onbekend voor het model. Daarnaast kan het doeleinde een ander jargon bevatten wat leidt tot out-of-vocabulary fouten.(Alharbi e.a., 2021)

Natuurlijke taalverwerking

Zoals vermeld in de vorige paragraaf is de use case van groot belang. De taal die gebruikt wordt verschilt op vele manieren: jongeren gebruiken andere woorden dan hun senioren, iemand in de IT sector heeft een andere woordenschat dan een dokter. Deze verschillen kunnen leiden tot out-of-vocabulary fouten. Dit houd in dat een bepaald model niet getraind is op bepaalde manier van spreken of woor-

[4] https://vocalremover.org/nl/

denschat. Een andere oorzaak van OOV is het trainen van een model op een te kleine dataset. (Alharbi e.a., 2021)
Ook niet iedereen spreekt woorden op dezelfde manier uit. Afhankelijk van de sprekers afkomst zullen ze een bepaalde taal met een bepaald dialect spreken. Ook wanneer ze in een taal anders dan hun moedertaal praten zal er vaak sporen van hun eigen taal herkenbaar blijven. Dit zorgt ervoor dat de uitspraak van een woord niet altijd hetzelfde is. (Alharbi e.a., 2021)

Efficiëntie van apparatuur
Als laatste hebben we de gebruikte toestellen. De kwaliteit van de opnameapparatuur kan de nauwkeurigheid van de ASR beïnvloeden. Het gebruikmaken van een high-end microfoon tegenover een goedkope kan betere resultaten creëren. (Alharbi e.a., 2021)

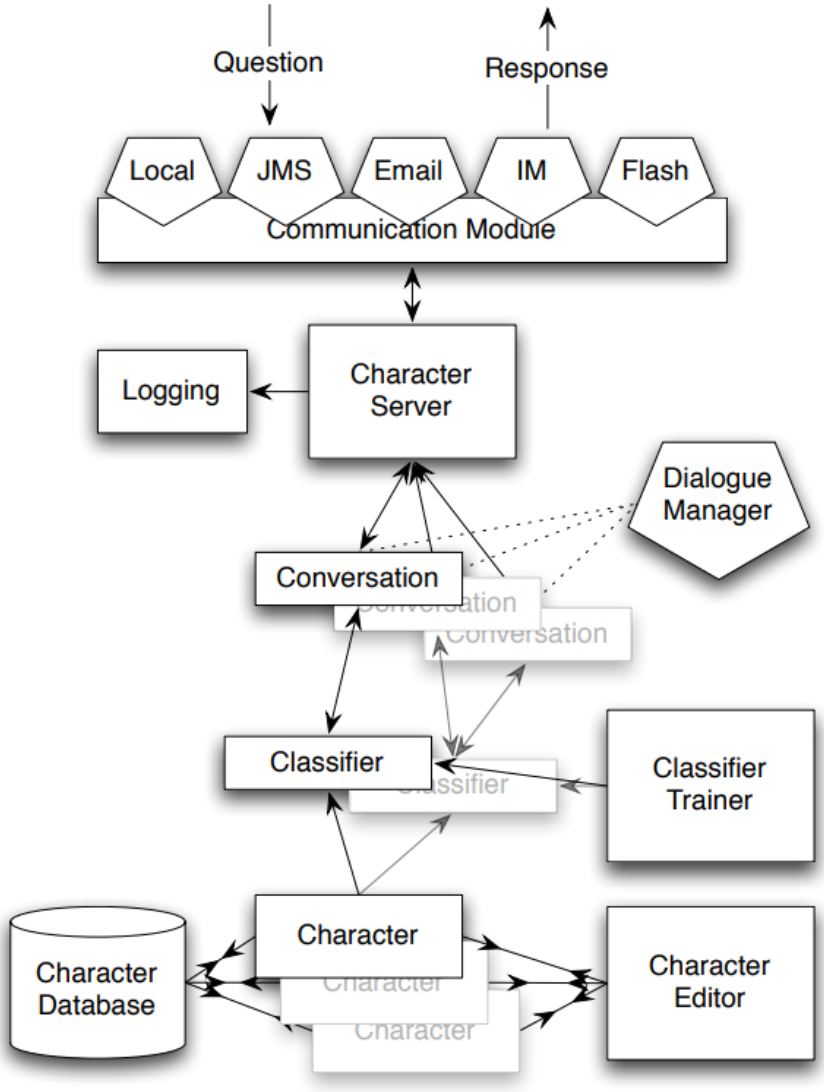
Sotteren
De moeilijkheden waarmee een ASR te maken krijgt bij stotter zijn als volgt: herhaling, verlengingen en blokkeringen (Manjula e.a., 2019). In het onderzoek van Suryaa en Vargheseb (2017) bespreken ze verschillende manieren om stotterfragmenten te transcriberen. Zo kan er een model getraind worden op vele stotterfragmenten om zo beter met stottermomenten om te gaan. Een andere aanpak is om vooraf de stottermomenten uit het audiofragment manueel of met andere software te verwijderen. Dat verwerkte fragment wordt daarna pas naar de ASR verstuurd.

2.3.2. Begrijpen van natuurlijke taal
Naast de ASR hebben we ook het begrijpen van natuurlijke taal (NLU). In 'Dimension in Testimony' maakten ze hiervoor gebruik van NCPEditor (Figuur 2.3). De classifier in dit systeem berekent welke antwoorden op de ingegeven tekst kunnen gegeven worden door de taalmodellen van beide te vergelijken en de antwoorden te rangschikken. Zolang de ingegeven tekst een foutmarge lager dan 50% blijft, zal het antwoord ongewijzigd blijven (Leuski & Traum, 2010).
Een andere bekend taalmodel is GPT-3. Dit model ligt aan de basis van de populaire chatbot ChatGPT.

2.4. Long list spraakherkening

2.4.1. Google Cloud Speech-to-text API
Beschrijving
Google Cloud Speech-to-text API[5] is Google's kijk op spraakherkennigssoftware. De API is heel betrouwbaar. Zo heeft het volgens Anggraini e.a. (2018) een succesper-

[5] https://cloud.google.com/speech-to-text



Figuur (2.3)
NPCEditor - Systeem Architectuur (Leuski & Traum, 2010)

centage van 100% voor normale Engelse stemmen en een percentage tussen 83,3% en 90% voor mensen met een spraakbeperking. Daarnaast biedt het ook de mogelijkheid om het model uit te breiden door het nieuwe vocabulaire aan te leren.

Functies

De service komt met verschillende functies. Dit zijn de belangrijke kenmerken die ze adverteren:

- **Taalherkenning**: De API ondersteunt spraakherkenning in verschillende talen en dialecten, waardoor gebruikers wereldwijd toegang hebben tot de dienst. Het kan automatisch de gesproken taal detecteren zonder voorafgaande taalconfiguratie.

- **Real-time streaming**: De API ondersteunt real-time streaming van spraak naar tekst, waardoor ontwikkelaars toepassingen kunnen bouwen die directe feedback en interactie vereisen, zoals live ondertiteling, spraakgestuurde commando's en real-time transcriberen.
- **Hoge nauwkeurigheid**: Dankzij de geavanceerde neurale netwerkmodellen levert de Speech-to-Text API nauwkeurige transcripties, zelfs in omgevingen met achtergrondgeluiden, verschillende spreekstijlen en spraak van meerdere sprekers.
- **Aanpasbare modellen**: Gebruikers kunnen de spraakherkenning aanpassen aan specifieke woordenschat en domeinen door aangepaste woordenlijsten en grammatica's te gebruiken. Dit verhoogt de nauwkeurigheid en zorgt voor een betere herkenning van vakspecifieke terminologie.
- **Multimediaondersteuning**: De API kan spraak herkennen in verschillende formaten, waaronder audio-opnamen, audiobestanden en streaming-audio. Dit maakt integratie met verschillende bronnen en toepassingen mogelijk.
- **Beveiliging en privacy**: Google Cloud hanteert strenge beveiligingsmaatregelen om de vertrouwelijkheid en integriteit van de gegevens te waarborgen. De API ondersteunt ook automatische verwijdering van persoonlijk identificeerbare informatie (PII) uit de gegenereerde transcripties.
- **Spraakherkenning op apparaat**: Naast de cloudgebaseerde API biedt Google Cloud ook een on-device spraakherkenning SDK aan, genaamd Speech-to-Text On-Device. Hiermee kunnen ontwikkelaars spraakherkenning rechtstreeks op apparaten uitvoeren, zonder een internetverbinding te vereisen.

Tarieven

De service kan per maand 60 minuten gratis gebruikt worden. Daarna moet er betaald worden per minuut van €0,016 voor standaard met datalogging tot €0,072 voor medisch gebruik zonder datalogging.

2.4.2. Amazon Transcribe

Beschrijving

Amazon Web Services (AWS) biedt een ASR software aan genaamd Amazon Transcribe[6].

Functies

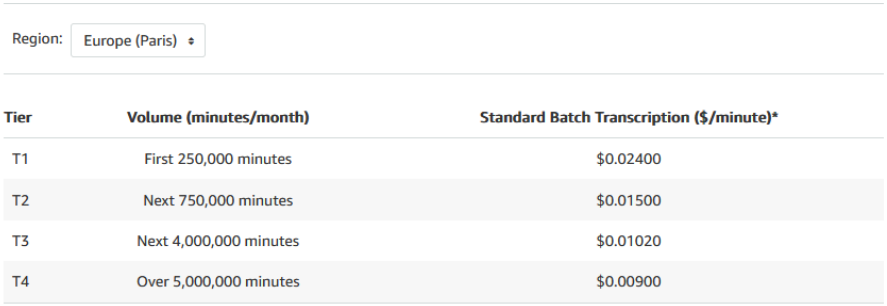
De functies die Amazon Transcribe aanbiedt zijn:

[6] https://aws.amazon.com/transcribe/

- **Taalherkenning**: Amazon Transcribe kan automatisch de gesproken taal detecteren, waardoor het mogelijk is om meertalige spraakherkenning te realiseren zonder voorafgaande taalconfiguratie.
- **Real-time streaming**: Amazon Transcribe heeft real-time streaming van spraak naar tekst, wat ideaal is voor toepassingen zoals live ondertiteling, spraakgestuurde opdrachten en interactieve dialoogsystemen.
- **Overzichtelijke transcripties**: Dit wordt gedaan door het toevoegen van tijdsaanduidingen en het herkennen van meerdere stemmen.
- **Aanpassing van modellen**: Gebruikers kunnen de spraakherkenning verbeteren en aanpassen aan specifieke vocabulaires en domeinen door aangepaste taalmodellen te maken. Hierdoor kunnen ze de nauwkeurigheid van de herkenning verhogen voor gespecialiseerde toepassingen.
- **Privacy**: Amazon Transcribe garandeert beveiligde dataverkeer. Daarnaast is het ook mogelijk woorden uit de transcriptie te filteren als deze ongewenst zijn.

Tarieven

De service kan 60 minuten per maand gratis gebruikt worden, gedurende 12 maanden. Daarna moet er een vergoeding worden betaald afhankelijk van het totale gebruik per maand en de geolocatie van de gebruikte server (Figuur 2.4).



Region: Europe (Paris)

Tier	Volume (minutes/month)	Standard Batch Transcription ($/minute)*
T1	First 250,000 minutes	$0.02400
T2	Next 750,000 minutes	$0.01500
T3	Next 4,000,000 minutes	$0.01020
T4	Over 5,000,000 minutes	$0.00900

Figuur (2.4)
Prijzen Amazon Transcribe voor Parijse server (Amazon, 2023)

2.4.3. Microsoft Azure Speech Services

Beschrijving

Microsoft Azure Speech Services[7] is een spraakherkenningsservice die wordt aangeboden door Microsoft Azure, het cloudplatform van Microsoft.

[7] https://azure.microsoft.com/en-us/products/ai-services/speech-to-text

Functies
Azure Speech Services biedt een breed scala aan functies voor spraakherkenning en tekst-naar-spraak-conversie. Enkele van de belangrijkste functies zijn:

- **Spraakherkenning**: De service kan gesproken taal detecteren en omzetten naar tekst in meer dan 60 talen. Het ondersteunt zowel real-time verwerking als batchverwerking van audiobestanden.
- **Taalherkenning**: Azure Speech Services kan automatisch de gesproken taal detecteren, waardoor het mogelijk is om meertalige spraakherkenning te realiseren zonder voorafgaande taalconfiguratie.
- **Aanpassing van modellen**: Gebruikers kunnen de spraakherkenning verbeteren en aanpassen aan specifieke vocabulaires en domeinen door aangepaste taalmodellen te maken. Hierdoor kunnen ze de nauwkeurigheid van de herkenning verhogen voor gespecialiseerde toepassingen.
- **Real-time streaming**: Azure Speech Services ondersteunt real-time streaming van spraak naar tekst, wat ideaal is voor toepassingen zoals live ondertiteling, spraakgestuurde opdrachten en interactieve dialoogsystemen.
- **Text-to-speech**: Naast spraakherkenning biedt de service ook mogelijkheden voor tekst-naar-spraak-conversie. Gebruikers kunnen tekst invoeren en de service genereert menselijke spraakuitvoer in verschillende stemmen en talen.

Tarieven
Per maand bied Azure vijf uur gratis aan daarna moet er een vergoeding van €0,935 betaald worden per uur.

2.4.4. Mozilla DeepSpeech

Beschrijving
In tegenstelling tot de voorgaande services is DeepSpeech van Mozilla open source.

Functies
Mozilla DeepSpeech[8] biedt verschillende functies en mogelijkheden voor spraakherkenning. Enkele van de belangrijkste kenmerken zijn:

- **Hoge nauwkeurigheid:** DeepSpeech streeft naar hoge nauwkeurigheid in spraakherkenning en heeft aanzienlijke vooruitgang geboekt in termen van herkenningsscores.

[8] https://deepspeech.readthedocs.io/en/r0.9/

- **Flexibiliteit:** Het systeem kan worden aangepast en afgestemd op specifieke toepassingen of vocabulaires. Hierdoor kan het worden gebruikt voor uiteenlopende spraakherkenningsscenario's.
- **Real-time verwerking:** DeepSpeech is ontworpen om spraak in realtime om te zetten, wat betekent dat het geschikt is voor toepassingen die onmiddellijke spraak-naar-tekstfunctionaliteit vereisen.
- **Privacy:** Als open-sourceproject stelt DeepSpeech gebruikers in staat om spraakherkenningssystemen lokaal uit te voeren, waardoor de privacy van gebruikersinformatie beter gewaarborgd is.

Tarieven
Aangezien dat de service open source is, moet er geen vast tarief betaald worden voor de service. Er kunnen echter wel kosten komen om de service te hosten.

2.4.5. Whisper

Beschrijving
Whisper[9] is de ASR service gemaakt door OpenAI. Het is beschikbaar als een API die ontwikkelaars kunnen integreren in hun eigen toepassingen en systemen door er spraakopnamen naar te sturen. De API geeft vervolgens een tekstuele weergave van de spraak terug als respons.

Functies
Whisper heeft verschillende kenmerken en mogelijkheden die het tot een krachtig spraakherkenningssysteem maken:

- **Hoge nauwkeurigheid:** Whisper heeft aanzienlijke vooruitgang geboekt in termen van spraakherkenningsscores en streeft naar hoge nauwkeurigheid bij het omzetten van spraak naar tekst.
- **Meertaligheid:** Het model van Whisper ondersteunt meerdere talen, waardoor het geschikt is voor wereldwijde toepassingen en gebruikers met verschillende taalvereisten.
- **Flexibiliteit:** Whisper kan worden aangepast en afgestemd op specifieke toepassingen en vocabulaires, waardoor het kan worden gebruikt in uiteenlopende spraakherkenningsscenario's.
- **Real-time verwerking:** Het systeem is ontworpen om spraak in realtime te verwerken, wat betekent dat het geschikt is voor toepassingen die onmiddellijke spraak-naar-tekstfunctionaliteit vereisen.

[9] https://openai.com/research/whisper

- **Privacy:** OpenAI hanteert strenge beveiligings- en privacyrichtlijnen om de gebruikersgegevens te beschermen. Whisper stelt gebruikers in staat om spraakherkenningssystemen lokaal uit te voeren, waardoor de privacy van gebruikersinformatie beter gewaarborgd is.

Tarieven
Net als Mozilla's DeepSpeech is Whisper open source. Dit betekent dat de kosten afhangen van waar en hoe de service wordt gehost.

2.4.6. Kaldi ASR

Beschrijving
Kaldi ASR[10] is een open-source spraakherkenningssysteem dat algemeen wordt gebruikt in de academische wereld en de IT industrie. Het is ontwikkeld door het Kaldi-project, een community-gedreven initiatief dat zich richt op het leveren van state-of-the-art spraaktechnologie.
Kaldi maakt gebruik van geavanceerde algoritmen en technieken, waaronder deep neural networks (DNN's) en hidden Markov models (HMM's), om spraak naar tekst om te zetten. Het biedt een flexibel en configureerbaar framework dat kan worden aangepast aan verschillende spraakherkenningstoepassingen.

Functies
Kaldi ASR biedt een breed scala aan functies en mogelijkheden die het tot een krachtig spraakherkenningssysteem maken:

- **Hoge nauwkeurigheid:** Kaldi maakt gebruik van geavanceerde modellen en trainingstechnieken om nauwkeurige resultaten te leveren.
- **Flexibiliteit:** Het Kaldi-framework is zeer configureerbaar en aanpasbaar. Het stelt gebruikers in staat om modellen en akoestische kenmerken aan te passen aan specifieke toepassingen en datasets.
- **Meertaligheid:** Kaldi ondersteunt meerdere talen en kan worden gebruikt voor spraakherkenning in verschillende taalomgevingen.
- **Uitgebreide toolkit:** Kaldi wordt geleverd met een uitgebreide toolkit die verschillende hulpmiddelen en utilities biedt voor spraakdataverwerking, modeltraining en evaluatie.
- **Community-ondersteuning:** Het Kaldi-project wordt ondersteund door een actieve gemeenschap van ontwikkelaars en onderzoekers, wat resulteert in regelmatige updates, bugfixes en nieuwe functies.

[10] https://kaldi-asr.org/

Gebruik en implementatie

Kaldi ASR wordt meestal gebruikt via de command line-interface en vereist enige technische kennis om effectief te kunnen gebruiken. Het proces om Kaldi te gebruiken omvat het verzamelen en voorbereiden van spraakgegevens, het trainen van akoestische en taalmodellen, en het uitvoeren van de spraakherkenning op nieuwe gegevens.
Kaldi biedt documentatie en handleidingen om gebruikers te begeleiden bij de implementatie en het gebruik van het systeem. Het heeft een actieve gebruikersgemeenschap waar gebruikers vragen kunnen stellen en ervaringen kunnen delen.

Beschikbaarheid en prijs

Kaldi ASR is een open-source project en is vrij beschikbaar voor iedereen. Het kan worden gedownload van de officiële Kaldi-website en lokaal worden uitgevoerd.

2.5. Long list natuurlijke taalverwerking

2.5.1. GPT series

Beschrijving

The GPT-series, wat staat voor Generative Pre-trained Transformer, zijn algemene taalmodellen gemaakt door OpenAI. Het doel van deze modellen is genereren van menselijke taal op basis van de gegeven input. Het meest bekende voorbeeld is OpenAI's chatbot, ChatGPT[11]. Het maakt gebruik van GPT-3.5 en GPT-4[12] om antwoorden te genereren. Deze modellen zijn getraind op een brede set van internet data.

Tarieven

Er zijn twee GPT-4 modellen beschikbaar de 8K context en de 32K context. Om ze te gebruiken moet er betaald worden per 1000 tokens. Dat is ongeveer gelijk aan 750 woorden. Voor de 8K context is de prijs $0.03 voor invoer en $0.06 voor de uitvoer. De kost voor 32K context is $0.06 voor invoer en $0.12 voor het resultaat. Afhankelijk van het model is de prijs voor GPT-3 tussen de $0.0004 of $0.02 per 1000 tokens.

2.5.2. BERT

Beschrijving

BERT[13], wat staat voor Bidirectional Encoder Representations from Transformers, is een ander type taalmodel ontwikkeld door Google AI Language. In tegenstelling tot de GPT-series, is BERT een bi-richtingstransformer, wat betekent dat het in staat is om zowel de context vóór als na een woord te begrijpen tijdens het trainen.

[11] https://openai.com/blog/chatgpt
[12] https://openai.com/gpt-4
[13] https://huggingface.co/blog/bert-101?text=Earth+can+be+saved+if+humans+[MASK]

Tarieven

In tegenstelling tot de GPT reeks is BERT gratis te gebruiken. Het taalmodel vereist wel aanzienlijke rekenkracht en geheugen waardoor de hardware waarop het draait sterk genoeg moet zijn.

2.5.3. FlairNLP

Beschrijving

FlairNLP[14] is een NLP (Natural Language Processing) framework ontwikkeld door Zalando Research. Het staat voor "Flexible Language-Agnostic IRst-orderën is gericht op het bieden van flexibiliteit en veelzijdigheid in het verwerken van natuurlijke taal. Net als BERT en GPT maakt FlairNLP gebruik van transformer-gebaseerde modellen en kan worden gebruikt voor een breed scala aan taalverwerkings-taken.

Tarieven

Ook FlairNLP is open-source wat wil zeggen dat de kosten afhangen van hoe en waar de software op draait.

2.5.4. RoBERTa

Beschrijving

RoBERTa[15] staat voor "A Robustly Optimized BERT Pretraining Approach."Het is een door Meta AI ontwikkelde transformer-gebaseerd taalmodel en een doorontwikkeling van BERT. RoBERTa is ontworpen om de prestaties van BERT verder te verbeteren door middel van optimalisaties in de trainingsprocedure.
In tegenstelling tot BERT maakt RoBERTa gebruik van een grotere hoeveelheid trainingsdata en een langere trainingsduur, wat het model helpt om een dieper taalbegrip te ontwikkelen. Het trainingsproces van RoBERTa omvat het maskeren van woorden in een zin en het uitdagen van het model om de ontbrekende woorden correct te voorspellen, vergelijkbaar met BERT.

Tarieven

RoBERTa is net als BERT gratis te gebruiken.

[14] https://github.com/flairNLP/flair
[15] https://ai.meta.com/blog/roberta-an-optimized-method-for-pretraining-self-supervised-nlp-systems/

3

Methodologie

Dit onderzoek start met een literatuurstudie (2). Vervolgens wordt er een requirementsanalyse gehouden waar gekeken wordt wat er van de oplossing verwacht wordt (3.1). Na de requirements vast te leggen zullen verschillende spraak naar tekst software (3.2) en taalverwerkingsmodellen getest worden. Als laatste zal er een proof-of-concept opgesteld worden.

3.1. Requirementsanalyse

De requirementsanalyse start met een opsomming van functionele en niet-functionele requirements door samen te zitten met iemand van het Zorglab. Vervolgens worden ze aan de hand van de MoSCoW methode gerangschikt op relevantie. Als laatste wordt er een use case opgesteld.

3.1.1. Functionele requirements

- De applicatie moet in staat zijn om spraak te herkennen en te registreren hoelang en tot wanneer de gebruiker spreekt.
- De applicatie moet in staat zijn om de gesproken woorden om te zetten in tekst, zodat deze kunnen worden opgeslagen en verwerkt.
- De applicatie moet in staat zijn om op basis van de gegenereerde tekst een volgend fragment te kiezen dat aansluit bij het onderwerp en de context van de tekst.
- De applicatie moet gebruik maken van AI-technieken om spraakherkenning en tekstgeneratie mogelijk te maken.
- De applicatie moet betrouwbaar zijn en weinig tot geen fouten maken bij het herkennen van spraak en genereren van tekst.

- De applicatie moet in staat zijn om de opgeslagen tekst te doorzoeken op bepaalde trefwoorden of zinnen, om zo snel specifieke informatie te kunnen vinden.

3.1.2. Niet-functionele requirements

NFR	Betrouwbaarheid
Indicator	Volwassenheid
Meetvoorschrift	De spraakherkenningssoftware moet een nauwkeurigheidsniveau van minimaal 95% bereiken bij het herkennen en transcriberen van spraak binnen de eerste 6 maanden na de lancering.
Norm	De spraakherkenningssoftware moet nauwkeurig genoeg zijn bij het herkennen en transcriberen van spraak, omdat eventuele fouten kunnen leiden tot onnauwkeurige resultaten en mogelijk onjuiste interpretaties.

NFR	Prestatie-efficïentie
Indicator	Snelheid
Meetvoorschrift	De spraakherkenningssoftware moet spraak in real-time verwerken met een maximale vertraging van 1 seconde tussen het uitspreken van de zin en het verschijnen van de transcriptie bij de lancering en deze snelheid behouden.
Norm	De software moet snel genoeg zijn om de spraak te herkennen en te transcriberen, zodat het in real-time kan werken en niet leidt tot onnodige vertragingen of onderbrekingen.

NFR	Beveiligbaarheid
Indicator	Vertrouwelijkheid
Meetvoorschrift	De spraakgegevens van gebruikers moeten worden versleuteld tijdens overdracht en opslag, en alle dataoverdracht en opslag moeten bij de lancering beveiligd zijn volgens de best practices, met regelmatige beveiligingsaudits.
Norm	De privacy en veiligheid van de gebruiker moeten worden beschermd, zodat de spraakgegevens niet kunnen worden gehackt of gelekt.

3.1.3. MoSCoW

Aan de hand van de MoSCoW methode wordt het belang van verschillende functionaliteiten bepaald.

Must have	De applicatie moet in staat zijn om de gesproken woorden om te zetten in tekst, zodat deze kunnen worden opgeslagen en verwerkt.
	De applicatie moet in staat zijn om op basis van de gegenereerde tekst een volgend fragment te kiezen dat aansluit bij het onderwerp en de context van de tekst.
	De applicatie moet gebruik maken van AI-technieken om spraakherkenning en tekstgeneratie mogelijk te maken.
	De applicatie moet in staat zijn om de opgeslagen tekst te doorzoeken op bepaalde trefwoorden of zinnen, om zo snel specifieke informatie te kunnen vinden.
Should have	De applicatie moet in staat zijn om spraak te herkennen en te registreren hoelang en tot wanneer de gebruiker spreekt.
	De applicatie moet betrouwbaar zijn en weinig tot geen fouten maken bij het herkennen van spraak.
Could have	De applicatie moet aanpasbaar zijn aan de specifieke wensen en behoeften van de gebruiker, zoals het leren herkennen van patronen in gebruikers met een spraakbeperking.
	Een flexibel systeem met verschillende scenario's creëren.

3.2. Spraakherkenning

Als eerste wordt er gekeken naar de spraakherkenning. Voor deze te testen wordt er gebruik gemaakt van fragmenten van het Nederlandse YouTube kanaal 'Stotter-Fonds' getest. Dit komt omdat door de algemene verordening gegevensbescherming (ook bekend onder de Engelse afkorting GDPR) er van audiofragmenten uit het zorglab jammer genoeg geen gebruik kan gemaakt worden. De woorden waar de spreker begint te stotteren staan in de fragmenten onderlijnd.
Voor elk model zal het resultaat, de tijd en de Word Error Rate (WER) worden berekend. In de resultaten staan de fouten van de ASR gemarkeerd. Deze fouten bestaan uit vervangen, invoegen en verwijderen van woorden. Op basis van deze fouten en het totale aantal woorden kan de WER berekent worden.

> WER = (Vervangingen + Invoegingen + Verwijderingen) / Totaal aantal woorden

3.2.1. Short list

Elk fragment zal door volgende modellen naar tekst worden omgezet:

- **Whisper's small model**
- **Whisper's medium model**

- **Gratis Google Cloud Speech API**
- **Microsoft Azure Speech Service**

Deze modellen zijn bedoeld voor het transcriberen van alledaags Nederlands.

3.2.2. Fragment 1 (Vloeiend)

'Hoi en wat leuk dat jullie kijken naar een nieuwe video. In deze video ga ik jullie iets vertellen over een moment waar ik in het dagelijks leven tegenaan loop wat betreft het stotteren.'

Whisper (small)

- **Transcriptie:** 'Hoi en wat leuk dat jullie kijken naar een nieuwe video. In deze video gaat jullie iets vertellen over het moment waar ik in het dagje sleef tegen aanloop. Wat betreft het stoptere?'
- **Tijd:** 1.98s
- **WER:** 23.52% =(7+0+1)/34

Whisper (medium)

- **Transcriptie:** 'Hoi! En wat leuk dat je kijkt naar een nieuwe video. In deze video ga ik jullie iets vertellen over een moment waar ik in het dagelijks leven tegen aanloop wat betreft het stotteren.'
- **Tijd:** 7.98s
- **WER:** 11.76% =(4+0+0)/34

Google

- **Transcriptie:** Hoi en wat leuk dat je weer kijkt naar een nieuwe video in deze video ga ik jullie iets vertellen over een moment waar ik in het dagelijks leven tegen aanloop wat betreft het stotteren
- **Tijd:** 4.21s
- **WER:** 14.71% =(4+1+0)/34

Azure

- **Transcriptie:** 'Hoi en wat lijken jullie kijkt naar nieuwe video in deze video ga ik jullie iets vertellen overal moment waar ik In het dagelijks leven tegenaan loop wat betreft het stotteren.'
- **Tijd:** 3.25s
- **WER:** 14.71% =(3+0+2)/34

3.2.3. Fragment 2 (Vloeiend)

'Ik was toevallig, van de zomer, was ik op een festival en ik wilde gewoon graag een broodje kroket, want dat vind ik gewoon super lekker.'

Whisper (small)

- **Transcriptie:** 'Het was een vallig van zomaar was ik op een festival en ik wil gewoon een beetje croquet, want dat vind ik gewoon super lekker.'
- **Tijd:** 0.98s
- **WER:** 34.61% =(6+1+2)/26

Whisper (medium)

- **Transcriptie:** 'Ik was toevallig van zomer op een festival en ik wilde gewoon een beetje kroket, want dat vind ik gewoon super lekker.'
- **Tijd:** 3.64s
- **WER:** 11.53% =(1+0+2)/26

Google

- **Transcriptie:** van de zomer was ik op een festival en ik wilde gewoon graag een broodje kroket want dat vind ik gewoon super lekker
- **Tijd:** 4.07s
- **WER:** 3.85% =(0+0+1)/26

Azure

- **Transcriptie:** 'Ik was toevallig van de zomer was ik op een festival en Ik wilde gewoon graag een broodje kroket, want dat vind ik gewoon super lekker.'
- **Tijd:** 1.48s
- **WER:** 0% =(0+0+0)/26

3.2.4. Fragment 3 (Lichte stotter)

'Als iemand aan mij ineens iets vraagt zeg maar en ik ben er niet op voorbereid, dan heb ik ineens. Dan sloeg ik dicht vaak en dan liep ik zomaar eens de klas uit.'

Whisper (small)

- **Transcriptie:** 'Als iemand ineens iets vraagt waar ik niet op voorbereid ben, dan heb ik ineens zo'n slug ik dicht vaak en dan liep ik zo meisje de klas uit.'
- **Tijd:** 1.21s
- **WER:** 17.64% =(4+0+2)/34

Whisper (medium)

- **Transcriptie:** 'Als iemand ineens iets vraagt en ik ben er niet op voorbereid, dan zoek ik dicht vaak en dan liep ik zo maar eens de klas uit.'
- **Tijd:** 3.81s
- **WER:** 2.94% =(1+0+0)/34

Google

- **Transcriptie:** ' ineens iets vraagt zeg maar en ik ben er niet op voorbereid dan dan heb ik ineens sloeg ik dicht vaak en dan liep ik zo maar eens de '
- **Tijd:** 6.60s
- **WER:** 17.64% =(0+1+5)/34

Azure

- **Transcriptie:** 'Zoals als, als iemand aan mij ineens iets vraagt, zeg maar, en Ik ben er niet op voorbereid dan. Dan heb ik ineens. Zo sloeg ik dicht vaak en dan liep ik zo maar eens de klas uit.'
- **Tijd:** 2.43s
- **WER:** 11.76% =(1+3+0)/34

3.2.5. Fragment 4 (Zware stotter)

'Ik heb eerst op logopedie gezeten. Maar ja, dat vond ik een beetje stom worden zo. Dus ik ben er vanaf gegaan. En toen ben ik vorig jaar, begin van de zomer, ben ik naar Ilanda gegaan. En nu zit ik ruim een jaar op therapie daar.'

Whisper (small)

- **Transcriptie:** 'Ik heb eerst op een logo paddie gezeten. Maar ja, dat vond ik een beetje stom worden zo. Dus ik ben gewoon afgegaan. En toen ben ik vorig jaar, begin van de zomer, ben ik naar die landaar gegaan. En nu zit ik ruim een jaar op terapie.'
- **Tijd:** 2.61s
- **WER:** 12.76% =(4+0+2)/47

Whisper (medium)

- **Transcriptie:** 'Ik heb eerst op Logo.de gezeten. Maar ja, dat vond ik een beetje stom worden. Dus ben ik gewoon afgegaan. En toen ben ik vorig jaar, begin van de zomer, ben ik naar Rilanda gegaan. En nu zit ik ruim een jaar op therapie.'

- **Tijd:** 8.21s
- **WER:** 10.63% =(3+0+2)/47

Google

- **Transcriptie:** 'Ik heb eerst op op logopedie gezeten maar ja dat vond ik een beetje stom worden enzo dus ben ik vanaf gegaan en toen ben ik vorig jaar zomer begin van de zomer ben ik naar Jolanda gegaan en nu en nu zit ik ruim een jaar op therapieën'
- **Tijd:** 15.14s
- **WER:** 19.14% =(3+4+2)/47

Azure

- **Transcriptie:** 'Ik heb eerst op op. Hallo logo die gezeten? Maar ja, dat vond ik een beetje stom worden zo, dus ben ik een van. Gewoon afgegaan, en toen ben ik voor vorig jaar zomer het begin van de zomer. Ben ik naar die landen gegaan? En nu het en nu zit ik die ruim een jaar. Therapie daar.'
- **Tijd:** 5.91s
- **WER:** 29.78% =(4+8+2)/47

3.2.6. Fragment 5 (Zware stotter)

'Soms wil je het grapje maken, weet je, maar als je stottert dan komt het een beetje vaag over. Dus ja, dan denk je: 'Oh, laat maar zitten.' Of met dan vrienden wil je wat zeggen, dat duurt zo lang en dan denk je: 'Ik zeg het maar niet meer.' Dat is soms wel jammer dat je dingen niet zegt vanwege je stotteren.'

Whisper (small)

- **Transcriptie:** 'Soms wil je het grapje maken, maar als je stottert dan komt het vaak over. Dus ik denk al laat me zitten. Of met vrienden wil je wat zeggen, dat doet zo lang . Ik zeg het maar niet meer. Dat is zo jammer dat je dingen niet zegt vanwege je stotteren.'
- **Tijd:** 2.72s
- **WER:** 20.63% =(6+0+7)/63

Whisper (medium)

- **Transcriptie:** 'Soms wil je het grapje maken, maar als je stottert komt het een beetje vaag over de zand. Dan denk ik, nou laat maar zitten. Of met vrienden wil je wat zeggen, dat duurt zo lang. Dan denk je, ik zeg het maar niet . Maar ja, dat is soms wel jammer. Dat je dingen niet zegt vanwege je stotteren.'

- **Tijd:** 9.57s
- **WER:** 11.11% =(1+4+2)/63

Google

- **Transcriptie:** 'soms dan Wil je het grapje maken als je stopt en dan komt het een beetje vaag over Dus ja dan denk ik oh laat maar zitten of met vrienden dan Wil je wat zeggen maar dat duurt zo lang en dan denk je oh ik zeg maar niet meer dat is soms wel jammer dat je dingen niet zegt vanwege je stotteren'
- **Tijd:** 19.02s
- **WER:** 6.34% =(2+1+1)/63

Azure

- **Transcriptie:** 'Soms, dan wil je het wel op de het de grapje maken, weet je, maar Als je stopt en dan komt dat een beetje vaag over. Dus ja, dan denk je nou laat maar zitten. Of met vrienden, dan wil je wat zeggen. Nou dat doet ze al lang en dan denk je, oh, ik zeg het maar niet, maar ja, Dat is zo zo. Zo is wel jammer dat je die dingen niet zegt vanwege je stotteren.'
- **Tijd:** 6.07s
- **WER:** 23.80% =(4+11+0)/63

3.3. Proof of concept

Als proof of concept wordt een python applicatie gemaakt. Deze applicatie biedt volgende functionaliteiten:

- Het opnemen van de microfoon
- Het transcriberen van audiofragmenten

Installatie

Voor dat de applicatie kan gebruikt worden moet er eerst aan een aantal dingen in orde gebracht worden. Eerst en vooral moet Python geïnstalleerd zijn. De applicatie is geschreven in versie 3.11.4. Nadat python in orde is kan je de applicatie klonen van Github met dit commando:

```
git clone https://github.com/P1kus3ru/ZorglAIb.git
```

Wanneer de applicatie is gedownload kunnen de afhankelijkheden geïnstalleerd worden. Open hiervoor een Command Prompt in de gedownloade map. Het pad zou moeten eindigen in '\ZorglAIb>'. Vervolgens kan met pip alle nodige pakketten geïnstalleerd worden:

```
pip install -r requirements.txt
```

Om Whisper te gebruiken moet Docker Compose[1] op het toestel worden geïnstalleerd. Dit is niet nodig voor de modellen die online draaien zoals Google en Azure. Deze hebben een geldige API key nodig.
Nadat de afhankelijkheden geïnstalleerd zijn moeten de environment variabelen opgesteld worden. Maak hiervoor een nieuw bestand aan in de 'src' folder en noem het '.env'. Daarna kan de inhoud van het bestand genaamd '.env.sample' worden gekopieerd naar het nieuwe '.env' bestand.

- **LOGGING:** Als dit op 'True' staat wordt er extra debugging informatie weergegeven in de console.
- **WHISPER_BASE_URL:** Dit is de poort waar Whisper kan bereikt worden.
- **TARGET_LANGUAGE:** De taal waarin er wordt gesproken.
- **REQUEST_TIMEOUT:** De tijd die de applicatie wacht op een antwoord anders geeft het een foutmelding.
- **MIC_RECORD_KEY:** De toets waarop geduwd moet worden om de microfoon op te nemen.
- **MICROPHONE_ID:** De id van het gewenste invoerapparaat.

De inhoud van het '.env.sample' kan behouden blijven behalve de variabele van de microfoon. Deze kun je vinden door het Python script 'get_audio_device_ids.py', dat zich bevind in 'src/modules', uit te voeren. Van de lijst die tevoorschijn komt kan dan het cijfer van het gewenste invoerapparaat aan de variabele worden meegegeven. Wanneer al het voorgaande in orde is gebracht is de applicatie klaar om gebruikt te worden.

Gebruik

Voor dat je de applicatie start, moet eerst Whisper klaargezet worden. Hiervoor moet in de 'ZorgIAIb' folder dit commando uitgevoerd worden:

```
docker-compose up -d
```

Daarna kan de applicatie worden uitgevoerd door het bestand 'voice_transcriber.py' uit te voeren. Als eerste zal het de gebruiker vragen met welk model het de audiofragmenten wil omzetten naar tekst. Momenteel zijn er drie mogelijkheden om van te kiezen: Whisper, Google en Azure. Vervolgens kan de gebruiker aan de hand van de gekozen toets op het toetsenbord iets inspreken. Daarna zal de applicatie dit via het gekozen model transcriberen en het resultaat samen met de duur weergeven in de console. De bedoeling was om vervolgens de resulterende tekst door

[1] https://docs.docker.com/desktop/install/windows-install/

te geven aan een NLC die dan een gepast fragment zoekt om op verder te werken. Dit laatste deel is niet binnen het termijn in gelukt.
Om alles weer af te sluiten kan in de Command Promt waarin het bestand draait de toetsen 'Ctrl+C' gedrukt worden. Dit stopt het Python bestand. Daarna kan ook Whisper gestopt worden door in de 'ZorglAIb' folder dit commando uit te voeren:

```
docker-compose down
```

4

Resultaten

Na de verschillende fragmenten door de modellen te laten gaan kunnen we uit de resultaten afleiden dat in Whisper's transcripties, zowel in het kleine als medium model, de boodschap van de gesproken fragmenten nog verstaanbaar is, ondanks fouten in het herkennen van sommige woorden. In vergelijking met het medium model is het kleine een stuk sneller, zo kan het kleinere model drie à vier keer sneller het fragment verwerken. Dit komt wel ten koste van de nauwkeurigheid aangezien het kleinere model meer woorden foutief herkent. De Whisper modellen blijven het meest consistent in hun nauwkeurigheid zo is te zien dat een stotter weinig impact heeft op het resultaat.
In tegenstelling tot Whisper duurt het maken van Google ASR's transcripties het langst. Zo neemt Google over het algemeen dubbel zo lang als Whisper's medium model met uitzondering tot het eerste fragment waar Google sneller is dan het medium model en het tweede fragment waar het het medium model bijna evenaart. Daarnaast is de nauwkeurigheid van Google's ASR beïnvloed door stottermomenten. Het presteert beter bij vloeiende spraak. Wat ook opvalt is dat Google's model de tekst allemaal aan elkaar hangt en zelf geen leestekens toevoegt aan de tekst.
In snelheid valt Microsoft Azure speech services tussen beide. Het model is sneller dan Google en Whisper's medium model maar trager dan het kleine model van Whisper. Bij vloeiende spraak gaf Azure nauwkeurige transcripties maar dat zakte wanneer het de fragmenten met meer stottermomenten verwerkte. In die fragmenten voegde het op de locaties waar de spreker stotterde dezelfde woorden meermaals toe.

5

Conclusie

Het onderzoek richtte zich op het gebruik van AI om stotterpatiënten een meer interactieve ervaring te bieden door spraakherkenningstechnologie toe te passen. De onderzoeksvragen waren gericht op het registreren van spraak, potentiële problemen bij spraakregistratie, de invloed van stotteraspecten op spraakherkenning en hoe de applicatie volgende spraakfragmenten kiest op basis van gegenereerde tekst.
Uit de literatuurstudie kwamen verschillende obstakels aan bod zoals domeinproblemen, natuurlijke taalverwerking problemen en de efficiëntie van de apparatuur. Aangezien de fragmenten bestonden uit video's van het Nederlandse YouTube kanaal 'StotterFonds', was er van achtergrond lawaai en meerdere sprekers geen probleem. Ook de taal en dialecten gaven geen moeilijkheid hoewel er geen duidelijkheid is hoe Belgisch Nederlands een verschil kan maken. Het domein waarin de modellen getraind zijn en de use case komen daarentegen niet overeen. De gebruikte modellen zijn gedoeld voor alledaags Nederlands en niet getraind op stotteren. De aanwezige obstakels bij de spraakherkenning van stottermomenten zijn vooral het herhalen van woorden en de blokkeringen. Deze zorgen ervoor dat de ASR de haperingen wilt registreren als aparte woorden of het woord meermaals achter elkaar zet. Dit was vooral op te vallen bij Azure.
De spraak wordt in de POC applicatie aan de hand van het gekozen invoerapparaat geregistreerd via een vooraf gekozen push-to-talk toets. Waarna het gecreëerde audiofragment met het gekozen model wordt verwerkt. Op basis van de resultaten blijkt het medium model van Whisper de beste optie voor het transcriberen van stotterpatiënten. Zo blijft de tijd en nauwkeurigheid van het model consistent over de fragmenten heen hoewel de duur wat langer is dan het Azure en kleine Whisper model. Die modellen hebben echter een hogere WER, vooral bij stotterfragmenten. Google is daarentegen uitgesloten dankzij de lange tijd die het neemt om een fragment te verwerken. Als laatste is het kiezen van een vervolg fragment

op basis van wat de spreker antwoord is niet tot stand gekomen binnen de tweede examenperiode.
De uitslag van dit onderzoek had ik niet verwacht. Zo veronderstelde ik op voorhand dat de betaalde services van Google of Microsoft de bovenhand zouden hebben tegenover de open source Whisper. Ik wist dat de open source community niet te onderschatten was met de vele bijdragers maar toch dacht ik dat de grote namen beter zouden presteren. Daarnaast ben ik teleurgesteld in mezelf en vind ik het jammer dat ik er niet in ben geslaagd om het kiezen van een vervolg fragment te realiseren.
Aangezien dit onderzoek was uitgevoerd in het kader van het 360° Zorglab, bieden de resultaten een meerwaarde. Hopelijk kunnen ze aan de hand van de cijfers en de POC verdere ideeën uitwerken om effectief toe te passen in hun stottertherapie of andere doeleinden. Ook kan deze paper als basis gebruikt worden om verder onderzoek naar spraakherkenning van stotterpatiënten uit te voeren of om het schakelen van fragmenten uit te werken en te implementeren. Zo kan er mogelijks onderzoek gedaan worden naar hoe het Zorglab zelf een model kan maken met de data die ze in hun sessies verzamelen.

A

Onderzoeksvoorstel

Het onderwerp van deze bachelorproef is gebaseerd op een onderzoeksvoorstel dat vooraf werd beoordeeld door de promotor. Dat voorstel is opgenomen in deze bijlage.

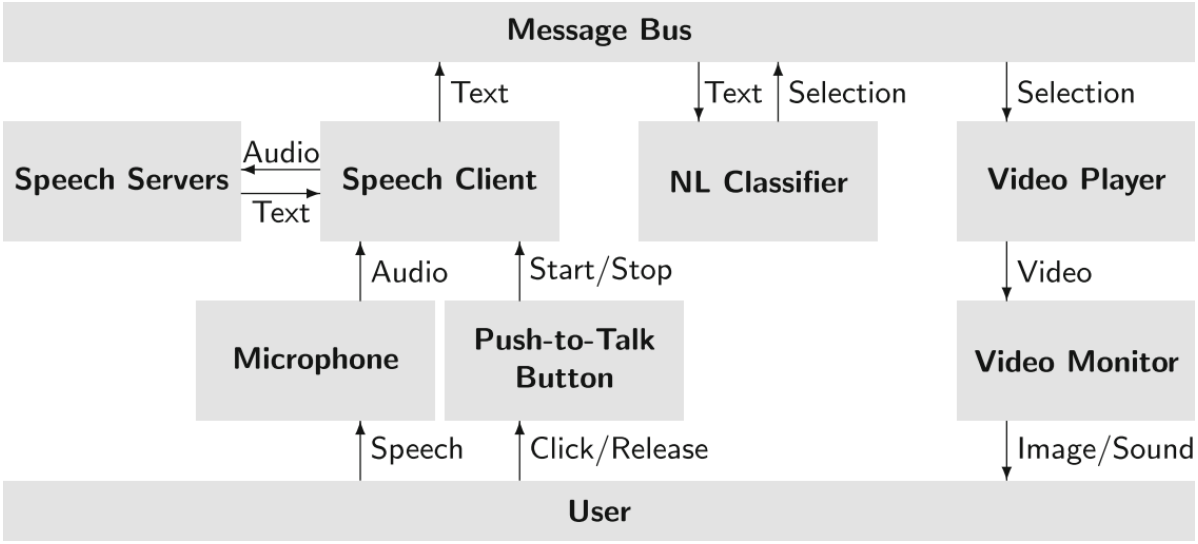
A.1. Introductie

In de reclamespot 'The Impact Will Be Real' over de Metaverse toont Meta (2022) hun visie over de rol die Virtual Reality (VR) speelt in de toekomst. Zo laten ze verschillende toepassingen ervan in het onderwijs zien. Jammer genoeg staat onze technologie nog niet zo ver als wat er in de reclame gezien kan worden maar ook nu al vindt VR zich een baan in verschillende opleidingen.
De Hogeschool van Gent maakt ook gebruik van VR om studenten de kans te geven in meer realistische situaties te oefenen. Hiervoor zijn twee verschillende technieken gebruikt. Allereerst heb je het renderen van een omgeving. Dit laat de gebruiker een interactieve wereld van 3D modellen ontdekken. Zo bestaan er drie virtuele kamers waarin de student kan oefenen. De andere manier is aan de hand van een 360° opname die wordt gemaakt aan de hand van een 360° camera. Omdat dit een opname van de werkelijkheid neemt, ziet deze methode er realistischer uit.
Dit is waar er op het probleem wordt gestoten. Aangezien de tweede methode werkt met een opname moet er naar verschillende fragmenten gesprongen worden naargelang het antwoord dat de gebruiker ingeeft. Dit wordt handmatig gedaan via een meerkeuzevraag. Hierdoor staat de oefening tijdelijk stil wat de echtheid van de situatie weghaalt. Daarom wordt in deze bachelorproef toegepaste informatica onderzocht hoe het overschakelen anders kan aangepakt worden zodat het voor de gebruiker realistischer aanvoelt. Hiervoor kijken we richting Artificiële Intelligentie (AI).

Alle grote IT bedrijven hebben wel een departement die zich bezighoudt met AI. Ze zien allemaal de mogelijkheden die het biedt. Van jobs gemakkelijker te maken tot het zelf creëren van kunst, de toepassingen zijn oneindig. Daarom zal in functie van het Zorglab op zoek worden gegaan naar een Spraak naar tekst (STT) en een natuurlijke taalverwerking software (NLP).
De STT zal worden gebruikt om het manueel ingeven van het antwoord te vervangen. In plaats daarvan zal de gebruiker gewoon zijn antwoord luidop kunnen geven en zal de STT dit in tekst omzetten. Dit alleen zal natuurlijk geen volgend fragment voor de gebruiker kunnen kiezen en daarom hebben we de NLP nodig. Deze zal de gegenereerde tekst omzetten naar kernwoorden en aan de hand daarvan bepalen welk fragment als volgende geschikt is.

A.2. State-of-the-art

Om de immersie in de simulatie te vergroten moet ervoor gezorgd worden dat het overschakelen naar een volgend fragment kan gebeuren zonder de gebruiker uit de illusie te halen. Hiervoor zouden verbale antwoorden gegeven kunnen worden. Een mooi voorbeeld van zo een interactie is het project 'Dimensions in Testimony' van de USC Shoah Foundation (2020). Daar kunnen bezoekers vragen stellen aan een overlevende van de Holocaust die op voorhand een volledig interview heeft afgelegd. Dit werd mogelijk gemaakt dankzij het systeem dat erachter zit (Figuur A.1). Dit systeem is opgebouwd uit volgende componenten: een software voor spraakherkenning (ASR) die de gebruikers verbale vraag in tekst omzet, een Natural Language Classifier (NLC) die op basis van de gegenereerde tekst een antwoord via een audio/video fragment voorziet en een mediaspeler die de fragmenten kan afspelen met tussendoor een inactieve animatie (Traum e.a., 2015).



Figuur (A.1)
Dimensions in Testimony - Systeem Architectuur (Traum e.a., 2015)

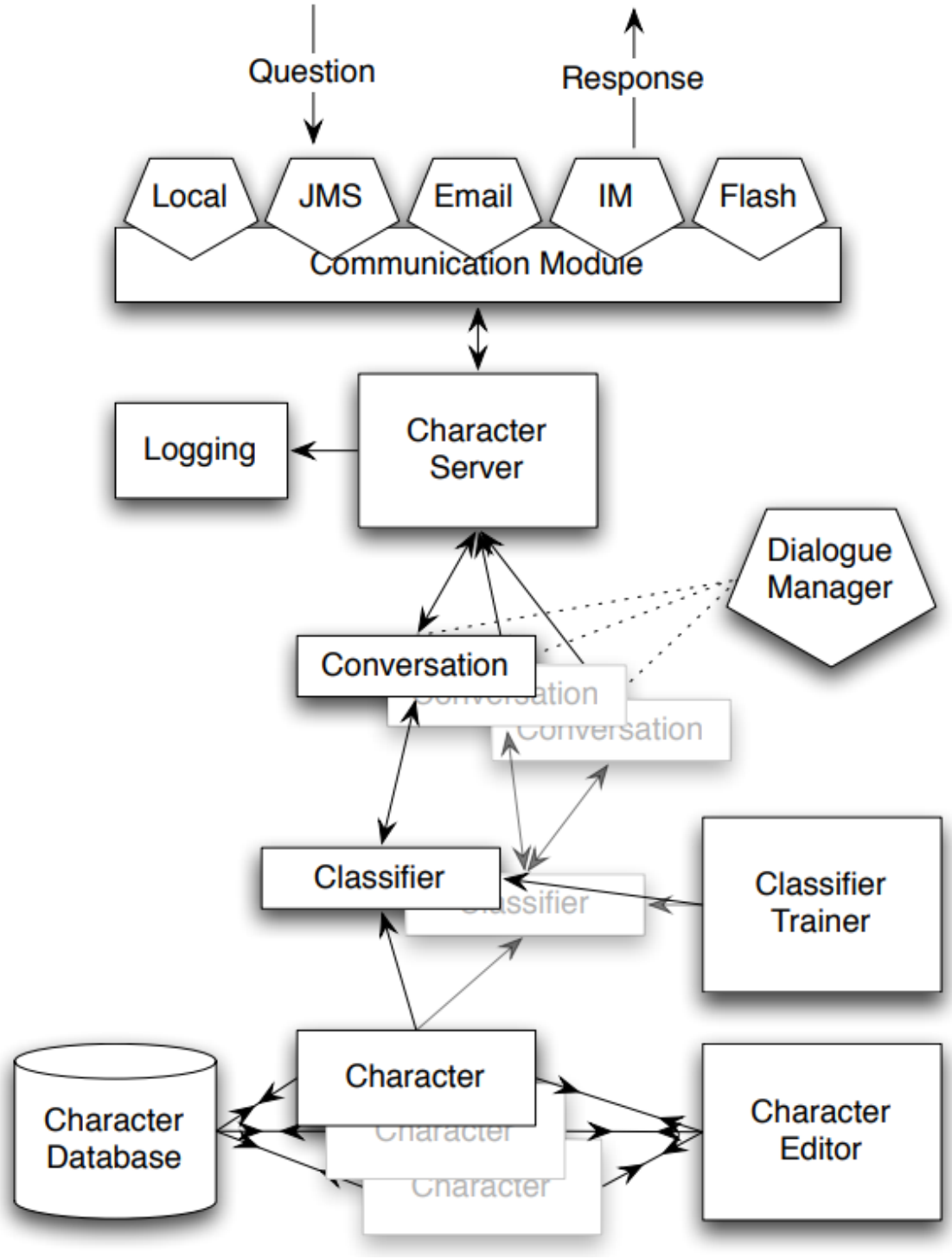
Spraakherkenning is de laatste jaren veel vooruitgegaan, dit komt omdat er meer data beschikbaar is, de algoritmes beter (deep neural networks) en de computers sneller zijn. Maar perfect zullen ze nooit zijn zegt van Hessen (2020). Dit is geen verrassing aangezien mensen zelf vaak nog problemen hebben bij het verstaan van een andere. De meest voorkomende fouten van een ASR zijn herkenningsfouten. Deze kunnen ontstaan door slechte kwaliteit van de opname maar ook van de manier waarop woorden worden uitgesproken. De andere fouten ontstaan omdat ASR zijn vocabulaire niet uitgebreid genoeg is. Dit noemen we Out-of-vocabulary-fouten (OOV) (van Hessen, 2020).
Naast de ASR hebben we ook de natuurlijke taalverwerking (NLP). In 'Dimension in Testimony' maakten ze hiervoor gebruik van NCPEditor (Figuur A.2). De classifier in dit systeem berekent welke antwoorden op de ingegeven tekst kunnen gegeven worden door de taalmodellen van beide te vergelijken en de antwoorden te rangschikken. Zolang de ingegeven tekst een foutmarge lager dan 50% blijft, zal het antwoord ongewijzigd blijven (Leuski & Traum, 2010).

A.3. Methodologie

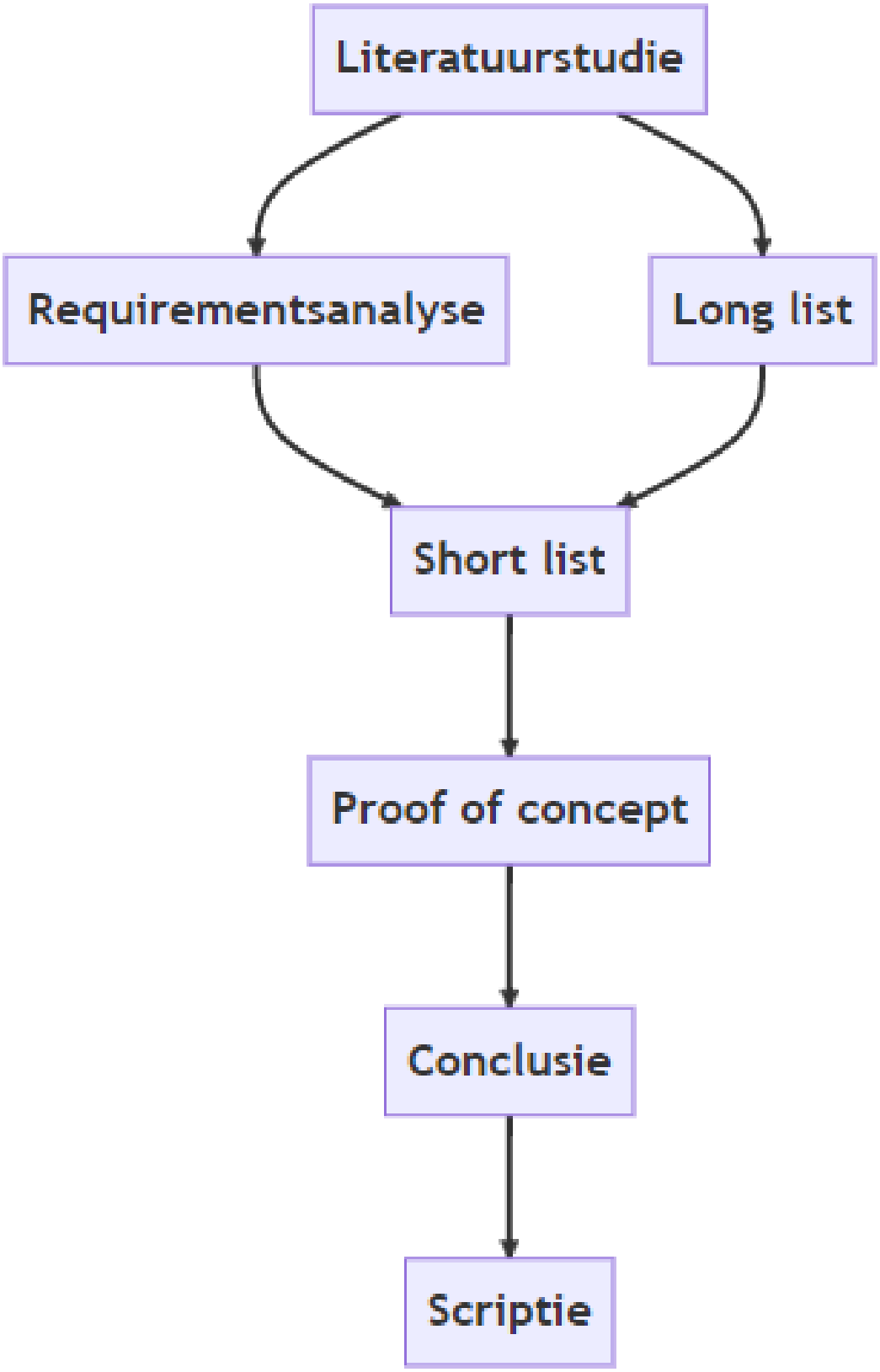
Om te beginnen zal er een uitgebreide literatuurstudie worden gehouden. Deze zal een overzicht bevatten van de vooruitgang en huidige stand van VR, ASR en NLP. Naar het einde van de literatuurstudie toe zal ook een long list worden opgesteld. Deze lijst zal bestaan uit bestaande ASR en NLP software. Voor elk item zal minimum een beschrijving, hun functies en tarieven bevatten. Ook zal er worden samengezeten met het team van het Zorglab om een requirementsanalyse uit te voeren. Zo kan worden neergeschreven wat de problemen zijn en waar ze tegenaan lopen. Daarin wordt besproken aan de hand van de MoSCoW methode wat er verwacht wordt van een oplossing, hoe moet deze in zijn werk gaan? Op basis daarvan zal de long list ingekort worden tot een short list worden opgesteld. Er zal bijvoorbeeld gekeken worden welke ASR's er al verkrijgbaar zijn en welke er in het Nederlands werken.
Wanneer er een uitgebreide literatuurstudie is uitgevoerd, zal een proof of concept opgesteld worden. Hierin wordt de software die gevonden is op de proef gesteld. Hier wordt gekeken hoe correct de ASR's inkomende audio kunnen transcriberen bij personen met een stotter. Daarnaast zal ook getest worden of er aan de hand van kernwoorden een juist fragment gekozen kan worden. Door de AI verschillende prompts te geven en te kijken of hij naar de juiste tijdsaanduiding in de video gaat. Nadat de poc afgewerkt is, wordt deze beoordeelt om te kijken hoe het in de context van het Zorglab toegepast kan worden.
Als laatste zal deze scriptie worden afgewerkt door het schrijven van een voorwoord, de samenvatting en een conclusie. Wanneer dit is voltooid zal deze proef worden beoordeeld waarna een positief resultaat een mondelinge verdediging zal plaatsvinden.

A.4. Verwacht resultaat, conclusie

Uit deze bachelorproef wordt verwacht dat er een proof of concept opgesteld wordt. Hierin zal gekeken zijn of het haalbaar is om aan de hand van spraakherkenning en natuurlijke taalverwerking software stottertherapie oefening, in het Zorglab, vlotter te doen verlopen. Zo kan het Zorglab kiezen of ze dit idee effectief aan het lab willen toevoegen. Dit kan dan aan de hand van de software dat in het PoC is gebruikt. Anderzijds kunnen ze op basis ervan andere software met hetzelfde doeleinde inzetten.



Figuur (A.2)
NPCEditor - Systeem Architectuur (Leuski & Traum, 2010)



Figuur (A.3)
Flowchart methodologie

Bibliografie

Alharbi, S., Alrazgan, M., Alrashed, A., Alnomasi, T., Almojel, R., Alharbi, R., Alharbi, S., Alturki, S., Alshehri, F., & Almojil, M. (2021). Automatic Speech Recognition: Systematic Literature Review. *IEEE Access*, *9*, 131858–131876. https://doi.org/10.1109/ACCESS.2021.3112535

Amazon. (2023, januari 1). *Amazon Transcribe: Developer Guide*. https://aws.amazon.com/transcribe

Anggraini, N., Kuniawan, A., Wardhani, L., & Hakiem, N. (2018). Speech Recognition Application for the Speech Impaired using the Android-based Google Cloud Speech API. *Telkomnika (Telecommunication Computing Electronics and Control)*, *16*, 2733–2739. https://doi.org/10.12928/TELKOMNIKA.v16i6.9638

Boas, Y. A. G. V. (2012). Overview of Virtual Reality Technologies.

Bryson, S. (2013). Virtual Reality: A Definition History - A Personal Essay.

Chee, L. S., Ai, O. C., Yaacob, S. B., & Perlis, J. (2009). Overview of Automatic Stuttering Recognition System.

Gordon, N. (2002). Stuttering: incidence and causes. *Developmental Medicine and Child Neurology*, *44*(4), 278–282. https://doi.org/10.1017/S0012162201002067

Leuski, A., & Traum, D. (2010). NPCEditor: A Tool for Building Question-Answering Characters. *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. Verkregen december 15, 2022, van http://www.lrec-conf.org/proceedings/lrec2010/pdf/660_Paper.pdf

Manjula, G., Shivakumar, M., & Geetha, Y. V. (2019). Adaptive Optimization Based Neural Network for Classification of Stuttered Speech. *Proceedings of the 3rd International Conference on Cryptography, Security and Privacy*, 93–98. https://doi.org/10.1145/3309074.3309113

Meta. (2022, juni 15). *The Impact Will Be Real*. Verkregen december 11, 2022, van https://www.youtube.com/watch?v=80IIEnSNwQc

Suryaa, A. A., & Vargheseb, S. M. (2017). Automatic Speech Recognition System for Stuttering Disabled Persons. https://api.semanticscholar.org/CorpusID:212570213

Traum, D., Jones, A., Hays, K., Maio, H., Alexander, O., Artstein, R., Debevec, P., Gainer, A., Georgila, K., Haase, K., Jungblut, K., Leuski, A., Smith, S., & Swartout, W. (2015). New Dimensions in Testimony: Digitally Preserving a Holocaust Survivor's Interactive Storytelling. In H. Schoenau-Fog, L. E. Bruni, S. Louchart & S.

Baceviciute (Red.), *Interactive Storytelling* (pp. 269–281). Springer International Publishing.

USC Shoah Foundation. (2020, april 3). *Dimensions in Testimony | USC Shoah Foundation*. Verkregen december 5, 2022, van https://www.youtube.com/watch?v=nGzAc9mIoTM

van Hessen, A. (2020, augustus 1). *Automatische spraakherkenning - Hoe kun je het inzetten voor onderwijsmateriaal?* (Onderzoeksrap.). SURF.