



Faculteit Bedrijf en Organisatie

Nursery tone monitor

Victor Standaert

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

Promotor:
Geert Van Boven
Co-promotor:
Jorrit Campens

Instelling: —

Academiejaar: 2020-2021

Tweede examenperiode

Faculteit Bedrijf en Organisatie

Nursery tone monitor

Victor Standaert

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

Promotor:
Geert Van Boven
Co-promotor:
Jorrit Campens

Instelling: —

Academiejaar: 2020-2021

Tweede examenperiode

Woord vooraf

Ik heb deze bachelorproef gekregen van mevrouw Teerlinck. Via haar kreeg ik het voorstel om deze bachelorproef te maken. Eerst vond ik het maar een vreemde opdracht maar toen ik er over nadacht wist ik dat ik met spraakherkenning en machine learning zou moeten gaan werken, en beiden interesseren mijn enorm. Bovendien is het een experimentele bachelorproef, nog niemand heeft zo een gelijkaardige opdracht gekregen. Dus wou ik de uitdaging wel aangaan.

Zelf wist ik niet goed wat te verwachten van deze bachelorproef. Ik wist dat het moeilijk ging zijn en dat ik mezelf veel zou moeten gaan aanleren. Want op school hebben we niks geleerd over spraaktechnologieën. Ik wist dat deze opdracht niet echt een einde had. Ik zou kunnen blijven onderzoeken en de app verder ontwikkelen of meerdere kenmerken van elderspeak aanpakken. Maar ik hoopte toch dat ik een werkend product zou kunnen afwerken en er een mooi onderzoek zou kunnen bijschrijven, wat me volgens mij wel gelukt is. Ik verwachtte wel dat ik veel zou bijleren. Dit moest ook om deze bachelorproef te voltooien natuurlijk. Maar ik ben ook blij dat deze verwachting is uitgekomen. Ik heb veel bijgeleerd over elderspeak, spraakherkenning, Python en een diepere kennis vergaard over machine learning.

Het was ook wel geestverruimend om te leren hoe groot rond probleem elderspeak wel is. Hiervan was ik totaal niet op de hoogte. Het was dan ook schokkend om te zien dat de mensen die hebben geantwoord op mijn enquête hetzelfde denken als alle studies die zeggen dat elderspeak zeker een probleem is. Op het einde van dit verhaal ben ik blij dat ik deze bachelorproef heb gekozen. Het was meer dan interessant om zelf al na te denken over de verschillende manieren om dit probleem aan te pakken.

Daarom wil ik als eerste graag mevrouw Teerlinck bedanken voor mij deze bachelorproef

aan te bieden. Zonder haar was dit avontuur nooit van start gegaan. Zelfs toen ik aan het twijfelen was om deze bachelorproef wel te nemen heeft ze me overtuigd om dit wel te doen. Hierbij wil ik dan ook de heer Campens bedanken voor mijn copromotor te zijn. Hij heeft me altijd enorm snel geholpen als ik met vragen zat en heeft me veel interne informatie over de verpleegkunde gegeven die me hebben geholpen bij mijn onderzoek.

Natuurlijk wil ik ook mijn promotor de heer Van Boven bedanken voor de goede feedback. Ook bedankt aan mevrouw Van Steenberghe die mijn vorige promotor was. Zij heeft mij ook goed op weg geholpen met altijd snelle antwoorden op mails en goede feedback.

Bedankt aan iedereen die mijn enquête heeft ingevuld. Zonder jullie had ik de onderzoeksdata niet die ik nodig had voor mijn onderzoek. En bedankt aan iedereen die vrijwillig mij wou helpen met audio in te spreken voor mijn applicatie. Bedankt aan mijn vriendin Joline. Zij studeert verpleegkunde en heeft me vaak uitleg kunnen geven over hoe elderspeak in de praktijk zich voordoet en hoe dit zou kunnen aangepakt worden.

Dank u wel iedereen, ik had het niet kunnen doen zonder jullie!

Samenvatting

In de sector van zorg- en verpleegkunde is elderspeak of de nursery tone een probleem dat al een tijdje aanwakkert. Ouderen of zwakkere doelgroepen worden niet graag betuttelend aangesproken. Dit komt vaak kleinerend over. De zorgkundigen weten dit maar vergeten het vaak omdat velen automatisch overschakelen naar zo een toon zonder dat dat hun bedoeling is. Daarom moet er onderzocht worden of het mogelijk is om deze nursery tone te kunnen herkennen met behulp van een applicatie. Staat de huidige technologie ver genoeg om deze vorm van communiceren te onderscheiden in een gesprek? Dit is het doel van deze bachelorproef. Er zijn verschillende kenmerken van elderspeak. En elk van deze kenmerken zal onderzocht worden of deze kan herkend worden door een applicatie.

In dit document zal u een literatuurstudie vinden met onderzoek over de verschillende kenmerken van elderspeak, waarom elderspeak juist een probleem is en in hoeverre dit probleem gevolgen met zich meedraagt, de verschillende technologieën die vandaag bestaan om spraak te herkennen, audio te analyseren... Er zal beslist worden welke technologieën de beste zijn om te gebruiken en ook wat de zorgverleners zelf vinden van dit initiatief. Uiteindelijk zal u zien hoe de applicatie is aangepakt en op welke manier de kenmerken van elderspeak herkend worden.

Het resultaat van dit onderzoek moet tonen of het mogelijk is om een bovenstaande applicatie te ontwikkelen. Het zal duidelijk blijken dat dit mogelijk is. Maar echter niet voor elk aspect van elderspeak evenveel. Sommige kenmerken zijn te herkennen: meer specifiek verkleinwoorden en herhalende zinnen. Andere zijn dan weer moeilijker, zoals de toonhoogte waarop iemand praat. Met een gespecialiseerd team die zich uitgebreid inzet op het ontwikkelen van een applicatie om elderspeak te herkennen, zou dit zeker mogelijk moeten zijn.

Het mag niet de bedoeling zijn dat de applicatie live feedback geeft. Want dit zou zorgen voor meer stress bij de zorgverleners en een storende inbreng in de kwaliteit van de zorg. Het zou eerder op het einde van een spreek sessie een overzicht kunnen geven met daarop de prestaties van de zorgverlener (dus hoe vaak of in welke mate men elderspeak heeft gebruikt).

Er zullen altijd nog open vragen blijven. Omwille van het feit dat de applicatie nooit 100% op punt zal staan. Er blijven altijd een aantal factoren die de precisie van de app kunnen dwarsbomen. Net zoals de spraaktechnologieën vandaag zijn er nog altijd momenten waar de software bepaalde woorden foutief verstaan. Dit wordt alleen maar beter in de toekomst.

Inhoudsopgave

1	Inleiding	15
1.1	Probleemstelling	16
1.2	Onderzoeksvraag	16
1.3	Onderzoeksdoelstelling	16
1.4	Opzet van deze bachelorproef	16
2	Stand van zaken	19
2.1	Verkenkend onderzoek	19
2.2	Onderzoek nursery tone	19
2.2.1	Elderspeak	19
2.2.2	Identificeren	20
2.2.3	Waarom elderspeak onderzoeken	21

2.3	Onderzoek technologieën	23
2.3.1	Gewenste technologieën	23
2.3.2	Beschikbare technologieën	23
2.3.3	Conclusie	30
3	Methodologie	31
3.1	Spraakherkenning	31
3.2	Verkleinwoorden	32
3.3	Toonhoogte	33
3.4	Herhalende zinnen	34
3.5	Plugins	35
3.6	Problemen	36
3.7	Emotie herkenning	36
4	Conclusie	39
4.1	Discussie	40
A	Onderzoeksvoorstel	41
A.1	Introductie	41
A.2	State-of-the-art	42
A.3	Methodologie	42
A.4	Verwachte resultaten	43
A.5	Verwachte conclusies	43

B	Bijlagen	45
B.1	Spraakherkenning	45
B.2	Verkleinwoorden	46
B.3	Frequentiemeter	47
B.4	Herhalende zinnen	48
B.5	Emotie herkenner	49
B.6	Enquête	51
	Bibliografie	57

Lijst van figuren

2.1	Results word recognition	26
2.2	Results speech in noise	27
2.3	Scaled amplitude plotted over 10 million samples	29
2.4	Power spectrogram (amplitude squared) for each mel over time .	30
3.1	Accuraatheid voorspelling testdata	38
B.1	Resultaat vraag 1	52
B.2	Resultaat vraag 2	52
B.3	Resultaat vraag 3	53
B.4	Resultaat vraag 4	53
B.5	Resultaat vraag 5	54
B.6	Resultaat vraag 6	54
B.7	Resultaat vraag 7	55
B.8	Resultaat vraag 8	55
B.9	Resultaat vraag 9	56
B.10	Resultaat vraag 10	56
B.11	Resultaat vraag 11	56

Lijst van tabellen

1. Inleiding

In onze maatschappij neemt het aantal ouderen gestaag toe. Dit wordt ook wel de vergrijzing genoemd. Tussen 2015 en 2050 zal de populatie van mensen boven de 60 jaar bijna verdubbelen van 12% naar 22% (Steverson, 2018). Hierbij komen zorgverleners dus meer en meer in contact met oudere zorgvragers. Het is dus een snel groeiende sector. Ook zorgverleners die in de thuiscontext aan de slag gaan, komen regelmatig in contact met ouderen, nu meer omdat steeds de nadruk komt te liggen op het “ageing in place”-beleid. Deze term wil zeggen dat de ouderen zo lang mogelijk in hun eigen woning wensen te verblijven, eerder dan beroep te doen op institutionele zorg.

Het is belangrijk dat de ouderen sociaal contact blijven houden. Er wordt zelf gezegd dat sociaal contact een beschermend effect kan hebben op de fysieke en mentale gezondheid van de oudere. Een eerder onderzoek (Gouveia, 2016) toont zelf aan dat het welzijn van ouderen meer wordt beïnvloed door een netwerk van vrienden dan door familie. Het hebben van meer dan één type relatie (namelijk vriendschap en familie) heeft een positieve impact op de kwaliteit van het leven van een oudere. Deze vriendschappen bevatten dus ook goede sociale contacten met zorgverleners.

Hier bevindt zich nu net het probleem. Niet enkel maken zorgverleners tijd vrij om met de ouderen een gesprek te hebben of een sociale band te creëren, maar er wordt ook vaak op een betuttelende of kinderlijke manier tegen hen gecommuniceerd. Deze communicatiestijl, die “elderspeak” wordt genoemd, kan men herkennen aan, onder andere, een overdreven toonhoogte (“nursery tone”).

De bedoeling van deze bachelorproef is om een applicatie te ontwikkelen die de communicatie tegenover ouderen gaat analyseren en de “nursery tone” kan onderscheiden van een normale toon. Eerst moet er natuurlijk onderzocht worden op welke manier en in welke

mate de gesprekken kunnen geanalyseerd worden.

1.1 Probleemstelling

Dit onderzoek is voor de doelgroep die met elderspeak aangesproken wordt en dit storend zou kunnen vinden. Dit zijn vooral oudere mensen in rusthuizen of ziekenhuizen, maar soms ook mensen van gemiddelde leeftijd. Vooral hiervoor zou deze applicatie een meerwaarde kunnen betekenen wetende dat deze groep het totaal niet aangenaam vindt om zo behandeld te worden.

Ook voor de zorg- en verpleegkundigen die de nursery tone of elderspeak gebruiken kan dit onderzoek van essentiële waarde zijn want meestal beseffen ze niet dat ze op die manier communiceren. Deze applicatie zou hun achteraf een signaal kunnen geven waardoor ze in de toekomst de nursery tone zouden kunnen vermijden. Deze applicatie is dus voor beide partijen voordelig.

1.2 Onderzoeksvraag

De onderzoeksvraag van deze bachelorproef luidt: 'Is het mogelijk om een applicatie te ontwikkelen die de nursery tone kan herkennen in een gesprek?'. We kunnen hier een aantal deelvragen aan koppelen. Namelijk:

- Hoelang zou het duren voor de applicatie volledig ontwikkeld is?
- Hoe accuraat kan deze applicatie de nursery tone herkennen in een gesprek?
- Hoe accuraat moet de applicatie zijn om in de praktijk te kunnen worden toegepast?

De deelvragen zijn belangrijk, maar niet zo belangrijk als de onderzoeksvraag zelf. Als deze met een 'Ja' kan beantwoord worden, is het een kwestie van tijd en testing om de applicatie volledig af te werken.

1.3 Onderzoeksdoelstelling

Het beoogde resultaat van deze bachelorproef is dat de proof-of-concept functioneel werkt. Het onderzoek en de literatuurstudie helpen met de kennis te vergaren die nodig is om deze proof-of-concept succesvol te maken. En enkel bij voltooiing binnen een bepaalde tijdsperiode, kan deze bachelorproef succesvol genoemd worden.

1.4 Opzet van deze bachelorproef

Het vervolg 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 wordt een 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

2.1 Verkennend onderzoek

De nursery tone wordt eerst onderzocht. Hierdoor wordt gehoopt meer kennis op te doen over wat de definitie van de nursery tone juist is. Want zonder deze informatie kan er niet aan verder onderzoek gedaan worden. Deze informatie is de fundering van het onderzoek.

Eveneens moet onderzocht worden wat de toon juist onderscheidt van een normaal gesprek. Zijn het de verkleinwoorden? Is het een hogere toon? Enzoverder. Dit zal helpen om te weten te komen in welke richting de applicatie verder zal ontwikkeld worden.

Als laatste onderzoek wordt er onderzocht of deze verschillen kunnen opgemeten of herkend worden aan de hand van een programma. En of er al technologie bestaat die kan gebruikt worden voor deze monitor.

2.2 Onderzoek nursery tone

2.2.1 Elderspeak

Elderspeak is een speciale manier waarop jongere volwassenen tegen ouderen communiceren. Deze manier is herkenbaar door eenvoudigere woordenschat en zin structuur, completere woorden, overduidelijke termen, lexicale woorden, duidelijke vragen met meerdere opties om op te antwoorden, de gezamenlijke “we” gebruiken, herhalen, trager articuleren, luider en op een hogere toonhoogte praten(Wikipedia, 2019).

Dit is het resultaat van het stereotiep dat ouderen minder goed kunnen horen, trager woordenschat kunnen verwerken en produceren. Ook al kunnen sommige aspecten van elderspeak voordelig zijn voor ouderen, wordt het meestal aangezien als ongepast, hinderend of storend in communicatie tussen verschillende generaties. Het is soms moeilijk om als zorgverlener te weten tegen wie ze normaal moeten praten en tegen wie op de manier van elderspeak. En wanneer bepaalde ouderen geen probleem hebben met deze toon wordt er soms overdreven met de elderspeak wat natuurlijk respectloos overkomt. Daarom zou een monitor deze vorm van elderspeak (of de nursery tone) kunnen onderscheiden en herkennen.

2.2.2 Identificeren

Hierboven hebben we de meeste kenmerken van elderspeak of de nursery tone. De applicatie die ontwikkeld zal worden zal deze tekenen proberen herkennen in een gesprek. Het geluid zal worden opgenomen in een ruimte waar iemand tegen een oudere persoon praat. Overdreven handgebaren of mondbewegingen zal niet kunnen herkend worden.

Er zal op een paar tekenen of kenmerken moeten gefocust worden en op een aantal andere minder. Want wanneer iemand luid praat kan de oudere ook echt een slecht gehoor hebben, terwijl er tegen niemand in de “we” vorm hoeft gepraat te worden. Dus niet elk kenmerk van elderspeak is even doorwegend om te beslissen of er hier weldegelijk in de nursery tone gesproken wordt.

Daarom zullen er een aantal kenmerken worden gekozen waar we diep onderzoek gaan naar doen en andere waar we weinig tijd aan gaan besteden. Zoals hierboven vermeld zullen we het volume waar iemand mee spreekt niet gaan onderzoeken. Dit is een duidelijk kenmerk van elderspeak en kan storend zijn voor de persoon tegen wie gepraat wordt maar het is niet mogelijk om voor elke patiënt een decibel-limiet in te stellen voor hoe goed die persoon hoort. Waarschijnlijk is een té luid volume voor de ene patiënt wel aanvaardbaar voor de andere omwille van het feit dat deze persoon minder goed hoort.

Een ander kenmerk die niet verder zal onderzocht worden is communicatie in de we-vorm. Zinnen zoals ‘We gaan dan eens gaan slapen.’ kloppen nu eenmaal niet want enkel de patiënt gaat slapen. Dit is dezelfde taal die we gebruiken tegen kinderen en is dus pure elderspeak. Maar ondanks dit kenmerk veel zou moeten laten doorwegen om te beslissen of er elderspeak wordt gebruikt of niet, is dit een moeilijk kenmerk om te laten herkennen door een applicatie. Want wanneer de we-vorm overbodig of correct gebruikt wordt valt niet te onderscheiden. De beide zinnen zijn hetzelfde en ook grammaticaal correct. Men zou de applicatie elke we-vorm kunnen laten herkennen maar dan weten wij en de verpleeg- of zorgkundige nog steeds niet of ze die we-vorm nu gewoon gebruikten op een correcte manier of als kenmerk van elderspeak.

Andere kenmerken zoals overduidelijke termen, completere woorden en betere articulatie zullen ook niet kunnen herkend worden. Dit simpelweg omdat de spraakherkenningstechnologie altijd woorden volledig herkent. Dus als iemand ‘Kheb et gedaan.’ zegt zal er ‘Ik heb het gedaan.’ worden herkend. Dus er kan hier geen onderscheid gemaakt worden

tussen completere woorden en betere articulatie. Voor overduidelijke termen zou men een woordenlijst kunnen opstellen die bestaat uit alle overduidelijke termen. Het is duidelijk dat dit een onbegonnen werk is en dat dit voor iedereen heel verschillend is. De ene persoon kan een term al meer overduidelijk vinden dan een ander.

De kenmerken die wel duidelijk te herkennen zijn, zijn verkleinwoorden en herhalende zinnen. Verkleinwoorden kunnen opgeteld worden. Zo kan de zorgverlener geconfronteerd worden met het aantal verkleinwoorden dat hij of zij heeft gebruikt. Voor herhalende zinnen geldt hetzelfde. Soms kan het zijn dat iemand echt een zin moet herhalen omdat hij het niet goed heeft verstaan. Maar het kan toch wel een handig overzicht geven voor de zorgkundige. Als laatste is de toonhoogte ook te herkennen. Hier kan veel over onderzocht worden, zowel in machine learning omgevingen als in andere. Hier wordt later nog op terug gekomen.

2.2.3 Waarom elderspeak onderzoeken

Het is bijna altijd onopzettelijk, maar niemand **vind** het leuk als iemand tegen je praat alsof je een kleuter of een gezinshond bent. En dan zeker niet door onbekenden. Het is niet enkel omdat de ouderen het “niet leuk” vinden. Neerbuigend praten tegen iemand is niet enkel irritant maar kan ook serieuze gezondheids- en gedrags- problemen met zich meebrengen.

Kristine Williams studeert elderspeak aan de universiteit van Kansas. “The underlying message of elderspeak is based on stereotypes that older people are less competent, it can cause injuries of self-esteem and make them more depressed so they avoid interaction.” (Abrahms, 2017) Volgens haar kunnen ouderen door elderspeak zichzelf aanzien als minder capabel. En dit kan leiden tot onder andere depressie.

Volgens nog een andere onderzoeker die de effecten van elderspeak bestudeert, gaat met elderspeak tegen iemand praten, een gevoel geven dat de patiënt onbekwaam zou zijn om iets te doen (Pfitzinger, 2019). Dit kan een soort kettingreactie in gang zetten die de oudere kan doen afdwalen naar depressie of een lager zelfvertrouwen. Mensen denken dan meer aan het feit dat ze met de jaren minder kunnen en beginnen daar over na te denken terwijl er vaak nog weinig of helemaal niks aan de hand is met hun zelfbekwaamheid. Deze zelfde onderzoeker zegt dat voor patiënten die lijden aan dementie de gevolgen van dit soort taalgebruik nog ernstiger kunnen zijn. Deze mensen kunnen zelf agressief of niet meewerkend worden.

Dit laatste wordt trouwens in verschillende andere onderzoeken ook geconfirmeerd. Bijvoorbeeld in onderzoek (Duffy, 2020) wordt duidelijk gemaakt dat mensen met dementie die te maken krijgen met elderspeak op ernstige negatieve manieren reageren. Soms schreeuwen, roepen, huilen... Dit is natuurlijk niet aangenaam voor de patiënt zelf maar de patiënt gaat zo ook de zorgverlening gaan verstoren. Hierdoor wordt het werk van de zorgverleners en verplegers onnodig moeilijker en zal de kwaliteit van de zorgverlening minder zijn.

Na interview met de heer Campens, lector verpleegkunde aan de hoge school HoGent, zien we dat de studenten les krijgen over elderspeak en dat ze over de symptomen leren, wat de gevolgen zijn en dat ze er op moet letten om elderspeak niet te gebruiken. **Waarom gebruiken zo veel mensen het toch nog? Wel, zoals eerder vermeld is dit niet opzettelijk. Het komt van een stereotiep dat ouderen niet goed meer zouden kunnen horen of niet bekwaam genoeg zijn.** En ook al bedoelen mensen daar geen kwaad mee, heeft het wel negatieve gevolgen en moet het aangepakt worden. (Simpson, 2014)

Genoeg redenen dus om de zorgsector te helpen met elderspeak.

Wat vinden mensen in de sector zelf?

Een vragenlijst die is ingevuld door ongeveer 100 mensen geeft veel confirmatie dat elderspeak een probleem is. Want maar liefst 92,8% van de mensen heeft 'Ja' geantwoord op de vraag 'Denkt u dat elderspeak een probleem is dat moet aangepakt worden?'. Dit is een overtuigend aantal zeker als men weet dat enkel 53,6% van de mensen had gehoord van elderspeak of de nursery tone voor ze deze enquête begonnen. Dus ook al wist men niet van deze term ze herkennen de beschrijving ervan wel van op de werkvloer. Want enkel 11,7% heeft geantwoord dat ze nog nooit in contact zijn gekomen met elderspeak bij hunzelf of anderen. Dit komt van zowel studenten als van zorgverleners die al lang in de sector werken.

In de vragenlijst werd ook nog bevraagd wat men van een nursery tone monitor vindt. De antwoorden waren verrijkend. **Volgens lector Campens zou de beste optie voor deze applicatie zijn dat deze live feedback geeft m.b.v. een seintje telkens de nursery tone wordt gebruikt.** Echter 27,8% antwoordt 'Nee' en 43,3% 'Misschien' op deze vraag. De opmerkingen hierbij zijn vooral dat men niet weet hoe dat in zijn werk zou gaan. Men vraagt zich af wat dat seintje juist gaat zijn, of het niet storend zou zijn? Een anoniem iemand geeft als opmerking: "De monitor geeft best niet onmiddellijk een voelbaar of hoorbaar sein bij gebruik van elderspeak, dit zou het gesprek teveel storen op het moment zelf." En deze persoon is niet de enige die er zo over denkt. We krijgen wel een compleet ander antwoord wanneer we vragen of men na een gesprek met een oudere een overzicht zouden willen met daarop feedback over hoe vaak of in welke mate ze de nursery tone hebben gebruikt tijdens dat gesprek. Hierop antwoordt 62,9% een 'Ja'. Dit is duidelijk. Men wil dus wel gebruik maken van de nursery tone monitor maar liever niet met live feedback. Een aantal mensen geven als opmerking dat dit overzicht achteraf een meerwaarde zou kunnen hebben en dat veel mensen hierdoor zouden leren elderspeak minder te gebruiken. En dit is exact waar we naar op zoek zijn!

Verder wordt nog bevraagd of men denkt dat de nursery tone monitor de strijd tegen elderspeak zou kunnen helpen en of men met zo een monitor zelf ook minder elderspeak zouden gebruiken. De antwoorden waren hierop beide eerder 'Ja'. De laatste vraag was of men denkt dat deze monitor meer stress zou geven aan het zorgpersoneel. En uit het onderzoek komt voor dat de meeste mensen ook denken van wel. Vooral dan als de uitkomsten van de monitor naar een leidinggevende zou gestuurd worden. Een bevroegde geeft als opmerking: "Of het al dan niet stresserend zou zijn voor de zorgverlener hangt

af van wie allemaal de feedback kan bekijken die de monitor heeft. Indien dit enkel voor de persoon die het bij zich heeft bekeken kan worden, denk ik dat dit als een 'leerschool' kan ervaren worden. Indien de feedback ook doorgestuurd wordt naar leidinggevenden of bijgehouden wordt voor andere doeleinden wordt het een ander verhaal. Ik geloof dat het een goed instrument zou zijn, indien het op een 'correcte' manier toegepast wordt met respect voor de zorgverlener zelf."

Uit deze antwoorden en opmerkingen lijkt het een goed idee om de resultaten van de app persoonlijk bij te houden zodat ze een leerschool kunnen zijn voor het zorgpersoneel zelf. Alle resultaten van de enquête kan u vinden op bijlage B6.

2.3 Onderzoek technologieën

2.3.1 Gewenste technologieën

Laten we eerst kijken welke technologieën we allemaal zouden willen gebruiken voor we kijken welke er beschikbaar zijn.

Aan de kenmerken van elderspeak te zien lijkt spraakherkenning (speech-to-tekst) een "must have" voor de applicatie. Hiermee zou er:

- Verkleinwoorden kunnen herkend worden die de woordenschat betuttelend maakt
- De gezamenlijke "we" vorm kunnen herkend worden als die te vaak gebruikt wordt
- Simpelere woorden en zinstructuur kunnen herkend worden
- Het type vragen kunnen herkend worden die worden gesteld
- ...

Een andere technologie die handig zou kunnen zijn is een soort toonhoogte (pitch) meter, en een volume (decibel) meter. Hiermee zou er:

- Het verschil in toonhoogte kunnen gemeten worden van een normaal gesprek
- Een zingende manier (op het einde van de zin een hogere toon) van spreken kunnen herkend worden
- Luider praten kunnen herkend worden
- ...

Misschien kan het ook handig zijn om te herkennen wie er praat aan de hand van hun stem. Zodat de applicatie weet of de oudere, de zorgverlener of een bezoeker praat.

2.3.2 Beschikbare technologieën

Spraakherkenning

Er zijn een aantal speech-to-tekst technologieën verkrijgbaar en zelf een paar daarvan compatibel met programmeertalen. De eerste die opviel en werd aangeraden was de google

cloud speech API. Deze ziet er gemakkelijk te gebruiken uit maar is helaas niet gratis.

Er zijn ook een paar open-source projecten online voor deze technologie. Hiervan zijn er wel een aantal outdated, en omdat spraakrecognitie altijd geüpdatet en verbeterd wordt zoek ik wel naar een recenter project.

We bekijken daarom **CMU Sphinx**. Een spraakherkenning toolbox voor mensen die applicaties willen ontwikkelen met spraakherkenning. Maar voor we deze technologie willen gebruiken, moeten we eerst wat meer te weten komen over hoe het werkt.

Spraakherkenning technologieën onderzoeken

Automatische spraakherkenning of automatic speech recognition (ASR) zie je tegenwoordig bijna overal. Meestal dienen ze om spraak in digitale tekst om te zetten. Zo kunnen mensen met computers communiceren via een ‘natuurlijke’ of menselijke manier. Voor deze bachelorproef zijn we op zoek naar een ASR technologie die speech kan omzetten in tekst maar ook een die een soort frequentie of pitch kan opmeten van de opgenomen audio. Deze moet betrouwbaar, accuraat en consistent zijn voor dit project om te kunnen werken. Want in een zorgomgeving kan er vaak met een dialect of accent gecommuniceerd worden en daar kunnen ASR systemen wel wat moeite mee ervaren.

Maar welke ASR systeem is nu het ‘beste’. De ‘beste’ is natuurlijk vaag. Wat er kan verstaan worden onder het beste ASR systeem is degene die van alle gesproken woorden er de meeste correct herkent en de minste foutief verstaat. De ASR systemen van grote bedrijven zijn hier sowieso het meest geschikt voor. Want deze zijn het meest ontwikkeld met de grootste budgetten. Denk aan die van Microsoft, Google, Apple, enzovoort.

Er is onderzoek naar gedaan **in 2017**. In dit onderzoek hebben ze de ASR systemen van Microsoft (Microsoft API), Google (Google API) en CMU Sphinx vergeleken. CMU Sphinx is geen ASR systeem van een groot bedrijf maar is wel het resultaat van 20 jaar van onderzoek van mensen uit de Carnegie Mellon University (CMU). Het onderzoek heeft getest op de ‘word error rate’ (WER), dus hoe vaak het ASR systeem het woord niet correct verstaat evenals op de accuraatheid. De resultaten waren 37% WER voor The Sphinx-4 (een technologie van CMU Sphinx), 9% WER voor de Google Speech API en 18% voor de Microsoft Speech API. Waarbij een lager percentage voor WER beter is (minder woorden foutief verstaan). Dus uit deze cijfers blijkt dat de Google Speech API superieur is (Bohouta, 2017). Maar dit onderzoek onderzoekt natuurlijk enkel maar de accuraatheid van de herkenning van Engelse woorden. Ze hebben wel 8 hoofd-dialecten van het Amerikaans Engels inbegrepen. Maar zal dit dezelfde uitkomst hebben voor onze Nederlandse dialecten?

Veel mensen denken dat ASR systemen niet in staat zijn op dialecten te begrijpen. Dit omdat er vaak mee wordt gelachen in filmpjes of sketches. Volgens een artikel over dialecten bij ASR systemen is dit immers onjuist (Lenke, 2017). Met behulp van machine learning en deep learning gebaseerd op neurale netwerken kunnen we een model trainen die wél in staat is om een dialect te leren begrijpen. Natuurlijk moet je dan wel het dialect reflecteren in je training data. En dit is niet zo vanzelfsprekend omdat er enorm veel

dialecten bestaan en ook tussendialecten die niet echt een naam hebben. Dit zelfde artikel wijst erop dat ASR systemen er enkel beter op worden om accenten te verstaan. Want door die machine learning kan de software zich aanpassen naar de toekomst. Hoe meer hij met het dialect of accent in contact komt hoe beter hij zal zijn om die te begrijpen. ASR kan dan nog niet alle dialecten en accenten volledig begrijpen maar het blijkt wel dat dit bij verschillende talen in stijgende lijnen toeneemt tot zelf 99 procent. Dit is bewezen door wereldwijde data vanuit verschillende ASR en NLU (natural-language understanding) gebruikt door duizenden apps in gsm's, auto's, IoT-toestellen (Internet of Things), enzovoort.

In het vakgebied van deze bachelor proef is dit wel van toepassing. De meeste bejaarde mensen hebben een zwaar dialect, een frans of ander accent. Velen praten geen Algemeen Nederlands. Volgens mevrouw Meerschaut, een stagiaire verpleegkunde klopt dit: “Ik moet vaak meerdere keren vragen om hun zin te herhalen omdat ik het gewoon niet versta. Ik vind dat wel irritant en ik denk de spreker zelf ook.”. Soms is het dus al moeilijk voor de zorgverleners zelf om de mensen te verstaan. Dus men kan zich inbeelden dat het voor een ASR systeem ook niet makkelijk is.

We moeten dus op zoek naar een ASR technologie die goed is in dialecten te leren of begrijpen. Omdat we nu weten dat de Google Speech API de beste keus is om te gebruiken als ASR systeem, gaan we kijken of het wel goed is met accenten en dialecten.

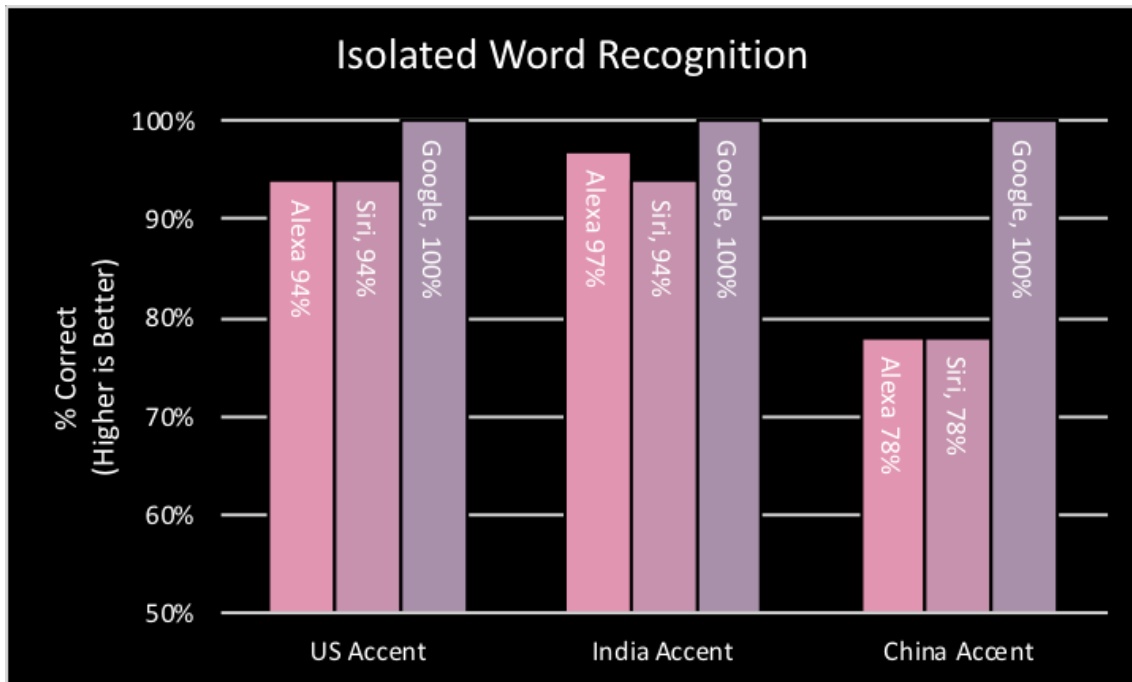
Een studie (Kinsella, 2018) in het lab van Vocalize.ai heeft onderzocht in hoeverre de ASR systemen van Amazon Alexa, Apple Siri en Google Assistant accenten kunnen begrijpen. Dit is een perfecte studie voor deze bachelor proef omdat in het vorige onderzoek de Google ASR al vergeleken werd met andere ASR systemen zoals Microsoft API, maar Amazon's Alexa en Siri van Apple zijn ook grote namen die worden aanschouwd als mede-marktleiders van virtuele assistenten met spraakherkenning. Het is dus goed dat we deze twee ook eens vergelijken met de Google speech API, ook wel bekend als Google Assistant. En de studie zelf is ook zeer relevant omdat we deze keer wél te maken hebben met sprekers met dialecten en/of accenten.

Voalize.ai stelde een schaal op van 1 tot 5 waarbij 5 geen accent is, dus iemand die Engels als zijn moedertaal heeft, en waarbij 1 een sterk en moeilijk te begrijpen accent is. Bij deze test waren er drie sprekers:

- Engels met een Amerikaans accent
- Engels met een Indisch accent
- Engels met een Chinees accent

Na het testen van 36 verschillende woorden is het resultaat te zien op figuur 2.1:

Hier zien we duidelijk dat de speech API van Google de winnaar is in het verstaan van accenten. Hij steekt er met kop en schouders bovenuit met een perfecte 100% accuraatheid op de 36 woorden voor alle accenten. Aan de hand van deze resultaten zou er al beslist kunnen worden dat er voor de Google speech API zou gegaan worden voor dit project. Vooral omdat in het vakgebied waar deze applicatie zal worden gebruikt er veel woorden met accenten zullen moeten herkend worden.



Figuur 2.1: Results word recognition
Source: Voicebot.ai

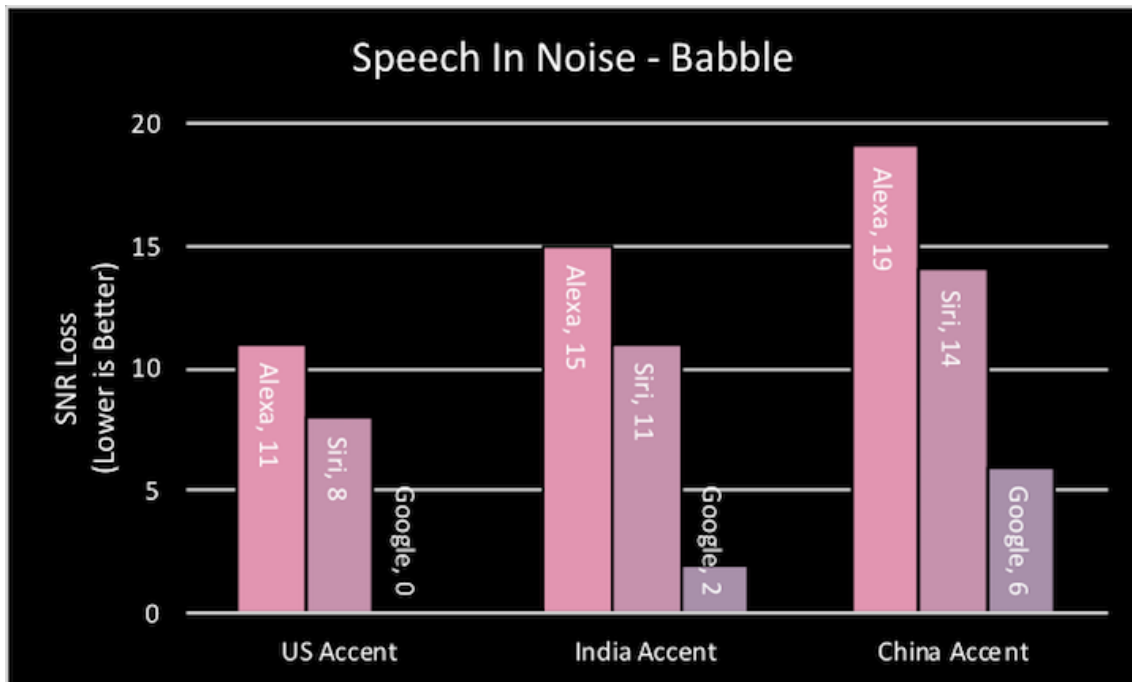
Maar Voicebot.ai heeft nog een test gedaan met de 3 ASR technologieën. Ze testen ook hoe goed de ASR presteert wanneer er achtergrondgeluid is, hier werd gebabbel gebruikt. Hier wordt er met een factor genaamd SIN (speech in noise) bepaald wie de bovenhand heeft. Een lagere score is beter. We zien in figuur 2.2 opnieuw de Google speech API winnen. Om er zelf nog een schepje bovenop te doen presteert Google ook beter op de test over verschillende volume levels. Zelf over de 3 accenten wint Google steeds.

Natuurlijk is hiermee de vergelijking nog niet gedaan met de CMUSphinx ASR. Deze is de enige die ook nog kan overwogen worden om te gebruiken doordat hij zo makkelijk te implementeren is in de code van dit project. Het is namelijk open-source, wat Google speech API niet is. Wat er wel gedaan kan worden is de Google ASR gebruiken voor de woorden (met accenten) te begrijpen, en CMUSphinx om de frequentiehoogte te meten. Want spraakherkenning en toonhoogte-herkenning is niet hetzelfde en zouden niet met elkaar gemengd mogen worden.

Plugins

Python is natuurlijk enkel een programmeertaal. Maar om een applicatie op de bouwen hebben we ook plugins nodig. Dit zijn extensies of librarys die meer mogelijkheden geven om mee te programmeren.

Er zijn een aantal plugins die interessant zijn voor deze applicatie. Dit zijn plugins speciaal voor Python. De eerste plugin die zeker zal moeten onderzocht worden is SpeechRecognition. Dit is een library voor zowel online als offline spraakherkenning. Deze library ondersteunt verschillende speech recognition API's zoals CMU Sphinx, Google Speech



Figuur 2.2: Results speech in noise
Source: Voicebot.ai

API, Microsoft Bing Voice Recognition en IBM Speech to Text. Hiervan zijn er een paar onderzocht in de literatuurstudie eerder in deze bachelorproef.

Tkinter is de volgende plugin die moet onderzocht worden. Dit is de standaard Python interface toolkit. Dit betekent dat alles van het uiterlijk van de app met deze plugin wordt gedaan. Hiermee kan er een 'window' geopend worden, labels en knoppen toegevoegd worden enzovoort. Het is wel niet enorm uitbreidbaar op gebied van uiterlijk. Dit betekent dat men met tkinter nooit een prijs zal winnen voor mooiste interface.

Een derde plugin die interessant lijkt is pickle. Pickle is een module die een Python data object kan omzetten in een byte stream. In andere woorden zorgt pickle voor het opslaan van data uit de Python applicatie in files op een computer en andersom. Dit kan van pas komen om gebruikers op te slaan. Mogelijks ook de gemiddelde snelheid waarop iemand praat om zo te beslissen of iemand sneller of trager praat dan gemiddeld. Als er per persoon metingen zullen worden gedaan kan het goed zijn dat er voor elke persoon een ander gemiddelde gaat zijn. De ene persoon praat wat hoger, luider of trager dan de andere en dit kan gevolgen hebben voor deze applicatie. Stel dat de frequentie wordt opgemeten van een gebruiker, dan kan er niet beslist worden of die persoon zijn frequentie hoog of laag is zonder dit te kunnen vergelijken met hun normale toonhoogte. Of iemand dus abnormaal praat is relatief ten opzichte van de persoon zelf.

PyAudio is nog een plugin die van belang kan zijn voor deze bachelorproef. Deze wordt gebruikt om audio af te spelen en op te nemen met Python op verschillende platformen en is ook nog eens makkelijk te gebruiken. Ook Wave is een interessante plugin. Wave is een module die instaat voor alle interactie met WAV files. Dit zijn audio files in het formaat

.wav. Wave kan deze openen, lezen, schrijven en nog een hoop andere dingen.

Nog een belangrijke plugin is NumPy. Dit is een package voor alles van wetenschappelijke berekeningen. NumPy is dus een library waarmee men bijvoorbeeld matrices, logische meetkunde, sorteer- en zoekalgoritmes, basis algebra, wiskundige statistiek en nog veel meer kan behandelen. De fundering van deze plugin is dat NumPy een object (een lijst) gebruikt die multidimensionaal is en geen data type heeft. Hierdoor kan er meer mee gedaan worden dan met een normale Python lijst. NumPy zou dus handig zijn voor de toonhoogte op te meten want hierbij komen wel een aantal wiskundige operaties bij te pas.

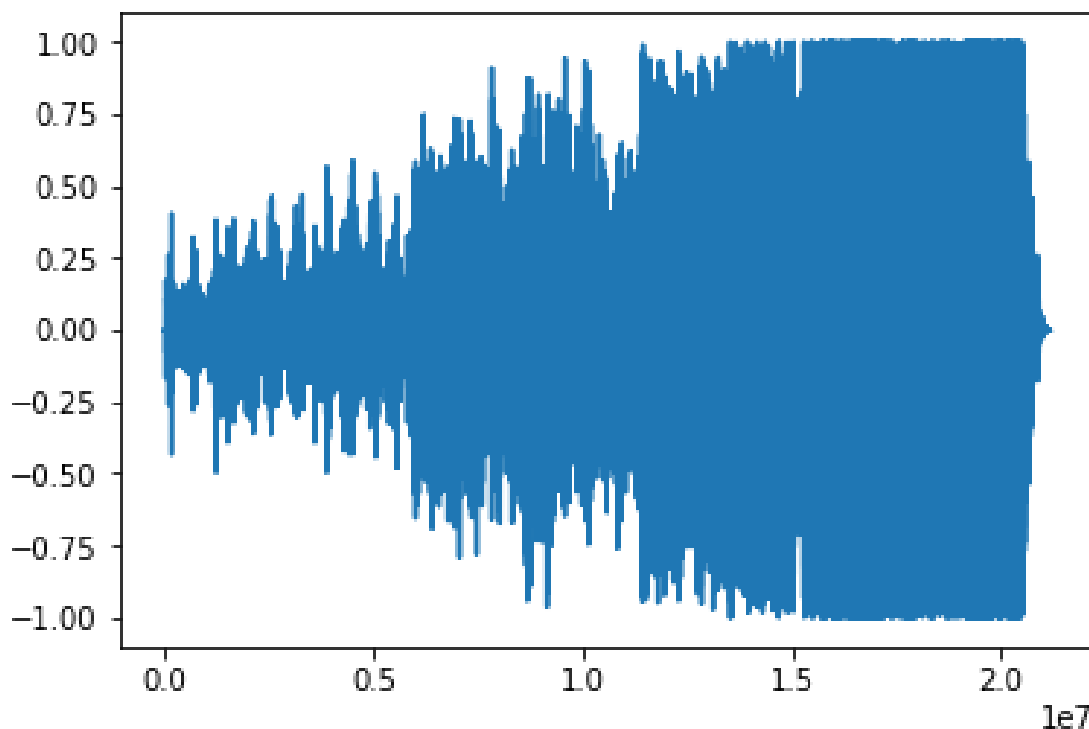
Stem herkenning

Er bestaat wel een probleem bij deze applicatie. Wanneer de nursery tone monitor in een ruimte zou gebracht worden of in een zorg- of verpleegkundige zijn of haar broekzak zit, zal deze ook altijd de stem van de patiënt opnemen en hetgeen wat die persoon zegt ook gaan analyseren. Dit is niet de bedoeling want het gaat enkel over de spraak van de zorg- of verpleegkundige. Als een patiënt verkleinwoorden uitspreekt zouden deze niet mogen mee opgeteld worden. Dit kunnen we enkel voorkomen door met stemherkenning te werken. Deze technologie gaat de stem van een bepaalde persoon leren herkennen en dan enkel op commando van die bepaalde persoon reageren. Dit zou dit probleem perfect kunnen oplossen.

De technologie bestaat al in verschillende ASR systemen zoals Google Assistant en Amazon Alexa. Maar is het ook mogelijk om dit in deze Python applicatie te implementeren? Het antwoord is ja. Dit is mogelijk met machine learning. En omdat deze technologie al bestaat zou deze van het internet in de applicatie kunnen worden toegevoegd. Het enige wat er nodig zou zijn is een dataset van gesproken audio met bij elke audio een label van de spreker. Dan zou er een trainingsmodel kunnen gemaakt worden die de eerste 75% van de data gaat gebruiken als trainingsdata. Daarna kan men de overige 25% gebruiken om het getrainde model te testen op accuraatheid. („Identifying speakers with voice recognition”, g.d.) Hierbij is de plugin librosa vooral van toepassing. Librosa is een Python package voor muziek en audio te analyseren. Het kan audio inladen en bijvoorbeeld een audiofragment visueel afbeelden in een diagram. Zoals afgebeeld in figuur 2.3.

Librosa kan ook de snelheid van een lied gaan opmeten in BPS (beats per minute) maar dit is niet bruikbaar voor deze applicatie omdat mensen niet in beats per minute praten. Wat wel handig is, is dat librosa ook een 'mel spectrogram' kan berekenen van de meegegeven audio. Eerst en vooral: Wat is een mel? Mel komt van 'melodic'. Een mel is een nummer dat overeenkomt met een pitch. Net zoals een frequentie overeenkomt met een pitch. Als men een noot pakt zoals A4, weten we dat die frequentie 440 Hz is. Als we een octaaf naar omhoog gaan wordt dat A5, wat overeenkomt met 880 Hz. Dus het dubbele. Als we nog eens omhoog gaan naar A6, verdubbelt de frequentie opnieuw naar 1760. Dus het verhoogt sneller per octaaf, ook al lijkt dit niet zo als mensen de verschillen tussen deze noten beluisteren. De mel schaal gaat deze intervallen van de frequenties gaan regulariseren.(Kaspar, 2019)

Gelukkige moet dit niet zelf ontwikkeld worden maar kan librosa dit doen met een inge-



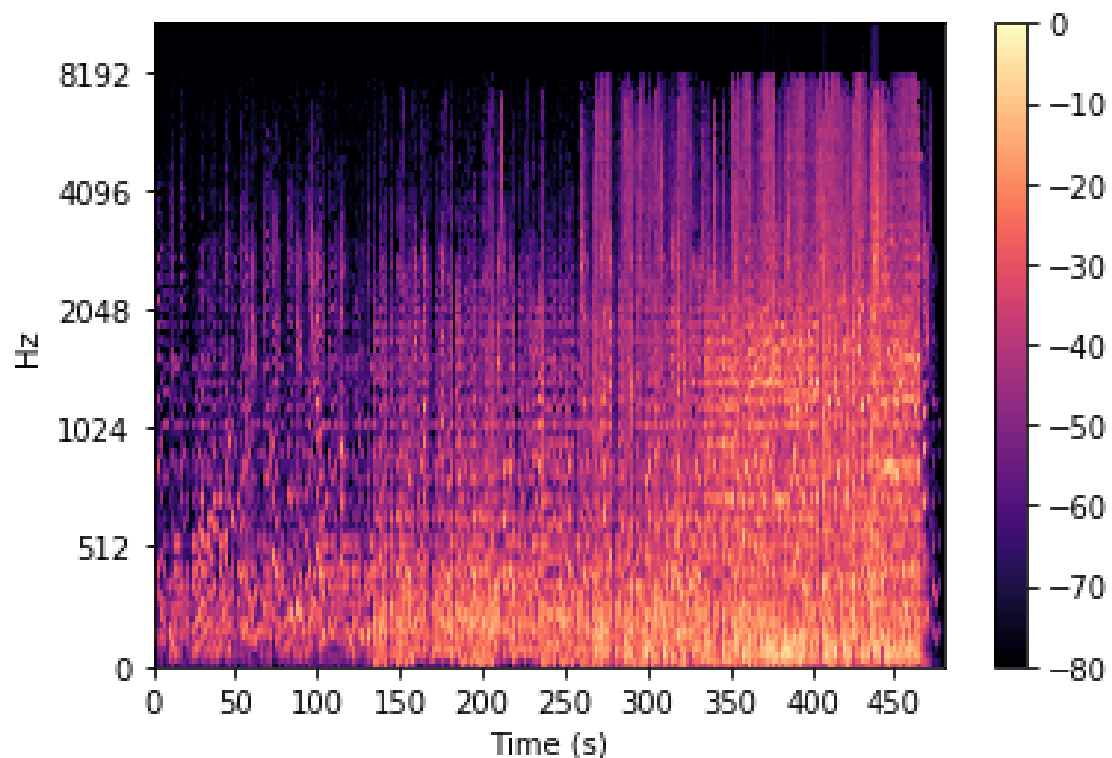
Figuur 2.3: Scaled amplitude plotted over 10 million samples
Source: Medium.com

bouwde functie. Deze geeft dan een spectrogram weer voor elke mel. Dit ziet er uit zoals op figuur 2.4.

Emotie herkenning

Een project die momenteel op het internet te vinden is, is een emotie herkenner. Dit is een open-source project (Livingstone, 2018)(„Python Mini Project – Speech Emotion Recognition with librosa”, 2021)). Dit project kan met behulp van machine learning ingesproken tekst onderverdelen in categorieën van emoties. Momenteel gebruikt dit project 8 emoties, namelijk: neutraal, kalm, blij, verdrietig, boos, angstig, walgend en verbaasd. Als dit model wordt getraind met 1440 ingesproken audio files krijgen we een accuraatheid van ongeveer 73%. Dit betekent dat dit project voor onbekende audio files kan voorspellen welke emotie er wordt gebruikt met een accuraatheid van 73%.

Nu zoeken wij voor onze bachelorproef geen emotie om te herkennen maar wel elderspeak. Er is niet meteen een emotie te plakken op elderspeak. Maar met de bestaande code kunnen we wel onze eigen versie maken van dit project. Want dit project gebruikt librosa, numpy en een neurale netwerk genaamd MLPClassifier. Deze drie zijn niet makkelijk te gebruiken maar we hadden deze al tegengekomen in dit onderzoek en het was al duidelijk dat we hiermee aan de slag gingen gaan.



Figuur 2.4: Power spectrogram (amplitude squared) for each mel over time
Source: Medium.com

2.3.3 Conclusie

Uit verschillende onderzoeken of studies waar men verschillende ASR systemen met elkaar vergeleken kwam de Google Speech API er altijd als beste uit. En dit in alle aspecten. Dus zal het eerste deel, die vooral over spraakherkenning gaat, deze technologie gebruiken. Voor de andere functionaliteiten van de applicatie zoals het herkennen van de toonhoogte zal er nog een andere technologie gebruikt worden. Deze API is makkelijk te implementeren in de applicatie in de programmeertaal Python, dus dit is dan ook de taal waar de applicatie zal in geschreven zijn.

3. Methodologie

Na het onderzoek naar de nursery tone of elderspeak, de kenmerken ervan, verschillende codetalen en de verschillende technologieën beginnen we aan de proof-of-concept. Deze wordt dus geschreven in de taal Python met voorlopig nog geen focus op een front-end (het uiterlijk/de interface van de applicatie). Dus de focus ligt op de functionaliteiten van de applicatie in orde te krijgen, zo kan er een antwoord gegeven worden op de onderzoeksvraag en dan later kan de code nog altijd in een mooier jasje gestoken worden, met als voorkeur een mobile omgeving. De keuze voor Python was snel gemaakt, de Google Speech API was vlot te implementeren in Python code. Maar deze taal is ook de koploper voor alles wat te maken heeft met machine learning en data analyse. Met in het achterhoofd dat er misschien wel machine learning zal moeten gebruikt worden om deze bachelorproef tot een goed eind te brengen én dat mobiele applicaties ook te schrijven zijn in Python, lijkt dit de beste keuze van taal.

3.1 Spraakherkenning

Nu moet er nog beslist worden hoe we aan de applicatie beginnen. Waar starten we? Wel, de kern van de applicatie zal spraakherkenning zijn. Met andere woorden: er moet eerst iets kunnen worden ingesproken die de applicatie dan kan begrijpen als gesproken tekst en die tekst dan digitaal uitschrijven. Dus als iemand 'Hallo test' zegt in een microfoon, moet de app 'Hallo test' kunnen uitschrijven als output (uitvoer). Als dit werkt kunnen we aan verschillende functionaliteiten beginnen. Dan kan er gekeken worden om de verkleinwoorden en de 'we'-vorm te gaan herkennen. Maar de spraakherkenning zelf is de fundering, dus hier zal de focus eerst op liggen. De code zelf kan u zien in bijlage B1.

Bij de simpele opname van spraak komen er een paar factoren kijken. Dit zijn namelijk de achtergrondgeluiden en de kwaliteit van de microfoon. Bij spraakherkenning is achtergrondgeluid niet zo een probleem. Tenzij dit achtergrondgeluid ook spraak is. Dan zou het kunnen dat dit gepraat in de achtergrond ook opgenomen en herkend wordt door de applicatie. Dit is natuurlijk niet de bedoeling. De kwaliteit van de microfoon speelt hier dezelfde rol. Wanneer de applicatie op een smartphone zou staan, zal de microfoon waarschijnlijk niet van de beste kwaliteit zijn. Dit kan er voor zorgen dat de spraak minder goed wordt herkend of dat er meer of minder achtergrondgeluiden worden opgenomen.

Dit probleem is ook moeilijk op te lossen. Want we moeten een bepaalde 'threshold' vinden. Een threshold is een grens van in dit geval decibels vanaf waar de applicatie gaat beginnen opnemen. Dit is nodig omdat er altijd decibels zijn in bijna elke ruimte. Ook al is het muisstil in een ruimte, zal er nog steeds 20-30 decibel zijn. En de applicatie moet voor deze decibels nog niet beginnen opnemen. Maar deze threshold instellen is niet vanzelfsprekend. Elke ruimte heeft namelijk een andere decibel gaande. En elke persoon praat dan ook nog eens op een verschillend volume. Het is dus niet mogelijk om 1 standaard threshold in te stellen voor alle gebruikers van deze app. Daarom moet hij dynamisch zijn. Dit kunnen we doen door de ruimte eerst te laten kalibreren voor dat de app begint op te nemen. Gelukkig bestaat er hiervoor een ingebouwde functie om dit te doen. Deze kan u ook zien in bijlage B1.

3.2 Verkleinwoorden

Verder moet er gekeken worden welk kenmerk van elderspeak we eerst gaan aanpakken. Het lijkt het best om de verkleinwoorden eerst uit de spraak te gaan filteren. Enerzijds omdat hier nog geen machine learning bij komt kijken. En anderzijds omdat dit bij elderspeak wel een zeer opvallend kenmerk is die veel zou doorwegen bij het beslissen of er nu in de nursery tone wordt gecommuniceerd of niet. De applicatie kan de woorden al begrijpen, dus nu moet er onderzocht worden wat de definitie juist is van een verkleinwoord.

Een verkleinwoord is namelijk een woord die altijd eindigt op of een -je, -tje, -mpje, -pje, -kje. Al deze woorden achtervoegsels hebben -je gemeen („Verkleinwoorden - hoofdregel | woordenlijst”, g.d.). Dus kunnen we simpelweg in de code de woorden één voor één doorlopen en controleren of ze op een -je eindigen of -jes voor het meervoud. Ook wordt er gecontroleerd op woorden die eindigen op -ke of -kes omdat er in het dialect veel verkleinwoorden met deze achtervoegsels worden gebruikt.

Maar dit gaat nog niet volledig kloppen. Er zijn namelijk ook woorden die op een -je of -ke eindigen die geen verkleinwoorden zijn. Dit zijn namen en woorden die kleiner zijn dan 3 letters. Er zijn een aantal namen die bijvoorbeeld op -ke eindigen maar dit zou niet als verkleinwoord moeten gezien worden. Bijvoorbeeld: 'Mieke'. De woorden met minder dan 3 letters zijn bijvoorbeeld gewoon 'je'. Dit is ook geen verkleinwoord maar eindigt wel op een 'je'. Dus moeten we ook controleren op de lengte van het woord en of het woord al dan niet overeenkomt met de namen van de verpleegkundigen en patiënten zoals u kunt zien in de code in bijlage B2.

Na verder onderzoek en testen blijkt dat er nog een hoop andere woorden zijn die eindigen op -ke. Namelijk bijna alle bijvoeglijke naamwoorden. Denk maar aan 'sterke' of 'leuke'. Maar omdat er niet direct een onderscheid te maken valt tussen bijvoeglijke naamwoorden die eindigen op -ke of verkleinwoorden die eindigen op -ke zal er gecontroleerd worden op alle andere woorden die eindigen op een -ke maar geen verkleinwoord zijn door een lijst van deze woorden te includeren in de applicatie. Deze komt van encyclopedy.nl (Encylo.nl, 2021). In deze lijst zijn ook alle steden, gemeenten en deelgemeenten gestoken die eindigen op -ke, simpelweg door een lijst te trekken van alle steden, gemeenten en deelgemeenten en te gaan kijken welke er eindigen op -ke. Want bijvoorbeeld de gemeente Merelbeke eindigt op -ke maar zou ook niet als verkleinwoord aanschouwd mogen worden. De lijst bestaat dus uit alle bijvoeglijke naamwoorden die eindigen op -ke, alle gemeentes die eindigen op -ke én ook nog een aantal uitzonderingen. Zoals 'ter zake' of 'ter sprake'.

Als het woord aan al deze voorwaarden voldoet en niet in de lijst voorkomt wordt dit woord toegevoegd aan een verzameling (of 'array' in programmeertaal). Op het einde van de opname van de spraak wordt er een overzicht getoond van alle verkleinwoorden die zijn uitgesproken en ook het totale aantal. Hierop kan dan een percentage worden berekend tegenover alle woorden die zijn uitgesproken. Zo kan je vergelijken of het aantal verkleinwoorden per normaal woord groter is dan in een gemiddeld gesprek, en zo beslissen of dit tekenen zijn van elderspeak. Het gemiddeld aantal verkleinwoorden per normaal woord is niet direct gekend. Dit zal moeten berekend worden door de applicatie. Want iedereen spreekt gemiddeld een andere aantal verkleinwoorden per zin (**avgdims**). Dus de grens vanaf wanneer er té veel verkleinwoorden gebruikt worden ligt bij iedereen anders. Dit kan mogelijks berekend worden door de applicatie een tijdje te laten luisteren naar hoe de gebruiker spreekt buiten de zorgsector en zo het gemiddelde te berekenen voor die persoon.

Verkleinwoorden zullen niet kunnen herkend worden via machine learning. Er is namelijk geen verschil tussen een normaal woord en een verkleinwoord dat een machine zou kunnen herkennen. Het enige verschil is dat het woord eindigt op -je of -ke. Maar daarvoor hebben we geen AI nodig. De machine zou zelf meer fouten maken doordat deze geen verschil zou zien tussen bijvoeglijke naamwoorden die eindigen op een -ke en verkleinwoorden die eindigen op een -ke. In andere aspecten dan spelling zou men kunnen denken dat een verkleinwoord misschien op een andere toon wordt uitgesproken, met een andere toonhoogte. Maar dan kunnen we beter direct naar de volledige herkenning gaan van toonhoogte in een zin, niet enkel in een woord.

3.3 Toonhoogte

Als volgende kenmerk van de nursery tone zal de toonhoogte of de pitch aangepakt worden. Vaak hoort men een verhoogde toon aan het einde van een zin of zelf in het midden van een zin van iemand die in de nursery tone spreekt. Als de applicatie de frequentie van een gesproken zin kan opmeten kan er gezien worden of er buitengewone pieken zijn. Het is mogelijk om in Python de frequentie of toonhoogte van audio weer te geven. In de applicatie is het mogelijk om de frequentie van een zin op te meten. Maar met deze

functionaliteit zal er niet veel vooruitgang geboekt worden om twee redenen:

Ten eerste is het moeilijk om ingesproken audio zuiver te krijgen, dit betekent dat er altijd ruis of achtergrond geluid bij is, en deze hebben ook frequenties. Hierdoor kan er niet gezien worden of een frequentiepiek door achtergrondgeluid komt of door de sprekende persoon.

Ten tweede is deze technologie die frequenties opmeet vooral bedoeld voor vaste tonen die geïsoleerd zijn. Bij spraak gaan mensen niet een aantal tonen perfect aanhouden maar zitten er ook veel andere geluiden behalve tonen; zoals medeklinkers. Deze zijn niet geproduceerd door de stembanden maar door luchtblokkering, lucht die door de keel of mond passeert of door de tong of lippen. Klinkers en medeklinkers hebben een andere frequentie. De gemiddelde pitch voor mannen ligt rond de 100-120 Hz, voor vrouwen en kinderen ligt dit rond de 300 Hz. Maar als we kijken naar die van medeklinkers gaan we zien dat deze rond de 500 Hz liggen (Microphones, 2021). Wat betekent dit concreet voor dit project? In de lijst of grafiek waar alle frequenties staan van een uitgesproken zin gaan er een aantal pieken in zitten. Dit kan enerzijds een hoge noot zijn die we zouden moeten beschouwen als kenmerk van elderspeak of anderzijds is het gewoon een medeklinker. Dus er is niet echt een manier om deze twee van elkaar te onderscheiden. Bijkomend is er dan ook nog eens het ruis en achtergrondgeluid die ook pieken zal geven. Er is bewezen dat bij spraakopnames voor toonhoogte-onderzoeken de achtergrondgeluiden steeds de grootste boosdoener was (Microphones, 2021).

De werkende code die frequenties opmeet kan u zien in bijlage B3

Ondanks dat het moeilijk of onmogelijk is om aan de hand van toonhoogte zelf uit te maken of het om elderspeak gaat of niet, betekent dit niet dat een machine dit niet zou kunnen. In de wereld van machine learning is er namelijk veel meer nodig dan men denkt. Een AI zou in deze frequentie sneller patronen kunnen zien en een model kunnen opstellen die elderspeak zou kunnen herkennen aan de hand van toonhoogte. (zie hoofdstuk 'emotie herkenner').

3.4 Herhalende zinnen

Een ander kenmerk die wel te doen is, is het herhalen van zinnen. Soms kan dit nodig zijn omdat de patiënt de zin niet goed verstaan heeft, maar vaak is dit ook gewoon een vorm van elderspeak. De manier waarop dit kenmerk zal aangepakt worden in de applicatie is met een soort cache. De applicatie zal een opgenomen zin opslaan in de cache voor een gelimiteerde tijd. Er zal dan gecontroleerd worden of de volgende zinnen overeenkomen met diegene in de cache. De cache zal per aantal zinnen die worden herkend leeggemaakt worden. Dit om de reden dat er wel zinnen kunnen terugkomen na een bepaalde tijd in het gesprek. En dit zou geen geval van elderspeak zijn want het gaat echt om het herhalen of herformuleren van net uitgesproken zinnen.

Er komt hier wel meer bij kijken in de programmeerwereld. Normaal zou men een zin kunnen onderscheiden met een punt om het einde van een zin aan te duiden of een

hoofdletter voor het begin. Maar sinds we met spraakherkenning werken krijgen we niet altijd een punt of een nieuwe hoofdletter. We krijgen enkel een hoofdletter bij de eerste zin die we uitspreken. Maar na de eerste zin worden de woorden gewoon herkend maar de spraakherkenning technologie kan geen onderscheid maken wanneer een nieuwe zin begint en een lange zin met een kleine stilte als pauze. Daarom zal het niet mogelijk zijn om een zin in de cache te steken maar zullen we moeten werken met de verschillende woorden en deze dan vergelijken met de komende woorden. Bij een groot aantal overeenkomende woorden kunnen we ervan uitgaan dat men de zin herhaald heeft of opnieuw geformuleerd heeft. Uiteindelijk komt het nog goed uit dat we geen volledige zinnen kunnen vergelijken met elkaar omdat een herformulering van een zin evenveel zou moeten meetellen als een volledig herhaalde zin.

Er moet ook nog onderscheid gemaakt worden met verschillende woorden die herhaald worden. Want de woorden 'ik', 'je', 'het', 'de', enzovoort, kunnen veel voorkomen zonder dat er echt sprake is van een herhaalde zin. De applicatie moet dus kijken of een combinatie van woorden vaak herhaald wordt en dat sommige van deze woorden ook langer zijn dan 3 letters.

De code om herhalende zinnen te herkennen kan u zien in bijlage B4.

3.5 Plugins

De applicatie maakt gebruik van een aantal plugins. Dit zijn plugins speciaal voor Python. Als eerste en belangrijkste plugin wordt SpeechRecognition gebruikt. Via deze library wordt er eigenlijk een bestaande API opgeroepen. Hierdoor hoeven we niet zelf een complete speech-to-text technologie te gaan programmeren maar kunnen we gewoon de Google Speech API oproepen in de app.

Voor de interface (het uiterlijk) van de python applicatie wordt Tkinter gebruikt. Ook al is het geweten dat tkinter niet de meest mooie interface aanbiedt. Hier is bewust voor gekozen omdat de applicatiefunctionaliteiten als prioriteit worden genomen. Als alles werkt kan mijn code nog altijd overgezet worden in een mooier beeld. Maar omdat deze bachelorproef niet gaat over een mooie app maken, wordt momenteel tKinter gebruikt.

Een derde plugin die de app gebruikt is pickle. In deze app worden de gebruikers opgeslagen via pickle in een bestand. Deze gegevens zijn ook geëncrypteerd. Deze files kunnen vervangen worden door een online databank. Er zal dan niet meer met pickle gewerkt worden. Maar omdat zo een databank geld kost is een lokale opslag via pickle momenteel een beter alternatief.

Voor de frequentiemeter die niet verder wordt gebruikt in deze app werden er ook een paar plugins gebruikt. PyAudio is er een van net als Wave. Over deze twee staat uitleg bij de literatuurstudie. Als laatste plugin wordt ook nog numpy gebruikt. Deze wordt gebruikt om de data door een wiskundige functie genaamd 'Fast Fourier Transform' (FFT) te laten gaan. Deze functie gaat de input (in dit geval de ingesproken audio) onderverdelen in componenten die op bepaalde frequenties zitten. Deze functie kan gebruikt worden voor

een groot aantal applicaties die digitale signalen verwerken. („Discrete Fourier Transform (numpy.fft)”, 2021)

3.6 Problemen

Bij de ontwikkeling van de applicatie komen er een aantal problemen kijken. Het belangrijkste probleem is dat voor sommige plugins te gebruiken men een versie van Python nodig heeft die men soms niet wil gebruiken. Bijvoorbeeld voor de stemherkenning en de emotie-herkenner te gebruiken hebben de de library librosa en numpy nodig. Librosa werkt namelijk niet in Python 3.7 maar wel in Python 3.6, 3.8 en hoger. En numpy werkt bijvoorbeeld enkel voor Python 3.7 of hoger. Dus er zijn een paar overlappende Python versies die we kunnen gebruiken om librosa en numpy in één applicatie te gebruiken; met name Python 3.8. Maar dit is dan niet compatibel met de plugins die het deel van de code die de verkleinwoorden gaat herkennen gebruikt. Deze werken enkel in Python 3.7.

Het komt er dus op neer dat we deze verschillende delen van de applicatie pas later zullen kunnen samenvoegen als de plugins gepubliceerd worden voor een gemeenschappelijke Python versie. Dit duurt normaal niet super lang.

Een tijdelijke oplossing voor de ontwikkeling van de applicatie is om environments te gebruiken. Zoals een Anaconda environment. Dit geeft een ontwikkelaar de kans om met een Python versie naar keuze te werken zonder zijn volledige computer om te zetten naar die Python versie.

3.7 Emotie herkenning

Omdat we nu een environment hebben in Python 3.8, kunnen we librosa gebruiken met onze andere plugins. Dit geeft ons de kans om aan de emotie-herkenner (deze is in de stand van zaken uitgelegd) te werken om deze om te vormen in onze eigen elderspeak herkenner. Dit met behulp van machine learning. De vorm van machine learning die we hier gebruiken is Feed-forward neurale netwerken (de basis vorm van neurale netwerken). De standaard machine learning systemen zoals SVM of Naive Bayes gebruiken dit niet. De voordelen van neurale netwerken te gebruiken is dat deze zichzelf dingen kunnen aanleren en ze kunnen een uitvoer geven die niet beperkt is tot de invoer die hun is gegeven. De invoerdata is ook opgeslagen in hun eigen netwerk in plaats van een databank. Dus al verliezen we data, zal het nog steeds blijven werken. (Thomas, 2020)

Om de audiofiles in te lezen gebruiken we soundfile. Soundfile gaat onze data openen en automatisch weer sluiten wanneer we er klaar mee zijn. Dit is makkelijk te gebruiken. In plaats van nu de bestaande 1440 audiofragmenten worden er andere fragmenten ingesproken die of normaal zijn of elderspeak zijn. Deze zullen dan net zoals de emoties gelabeld worden als normaal of elderspeak en dan ook in het model gestoken worden om de AI te trainen op deze data.

Dit project (de emotie herkenner) controleert de audio op 3 gebieden:

- mfcc: Dit is de Mel Frequency Cepstral Coefficient. Dit is de power spectrum op korte termijn van een audiofragment.
- chroma: Betreft de 12 verschillende pitch klassen. Zoals A, B, C#...
- mel: Mel Spectrogram Frequency

Al deze puntjes hebben eigenlijk betrekking tot de frequentie of pitch van het geluid. Dit komt goed uit, want de verkleinwoorden kunnen we al herkennen en herhalende zinnen ook. De toonhoogte werd dus nog niet aangekaart.

Dit project is dus omgevormd naar een elderspeak-herkenner in plaats van een emotie-herkenner. In plaats van de 8 emoties hebben we er nu maar 2, namelijk neutraal en elderspeak. Wat we dan nodig hebben is input data om ons model te trainen. We geven ons model dus ingesproken korte zinnen die ofwel in een neutrale toon zijn ingesproken ofwel in de nursery tone. Dit wordt gedaan door verschillende sprekers, waaronder een mannelijke en een paar vrouwelijke sprekers om de diversiteit er in te houden. Want mannen en vrouwen spreken op een andere toonhoogte dus moeten we het model ook aanleren dat in beide gevallen elderspeak kan voorkomen.

Bij machine learning geldt: hoe meer input data, hoe beter. Dit maakt het model namelijk preciezer. Maar om 1440 zinnen in te spreken hebben we misschien te weinig resources. Er worden dus 48 zinnen ingesproken per spreker, dat brengt ons totaal op 144. Hiervan zijn er 24 in een neutrale toon, 24 in elderspeak. En dit komt van 12 verschillende zinnen die 2 keer zijn ingesproken.

Wanneer we het model dan gaan trainen met 75% van onze input data, en dan het model de overige 25% laat voorspellen komt het uit op een accuraatheid van 72% zoals te zien op afbeelding 3.1. Dit betekent dus dat het model goed getraind is en nieuwe onbekende data juist kan labelen als neutrale spraak of elderspeak. Maar het is ook niet perfect getraind. 72% betekent nog altijd dat het model nog testdata als foutief labelt. Dit kan komen door de kwaliteit van de microfoon, achtergrondgeluid en medeklinkers die gebruikt worden zoals besproken bij de stand van zaken. Het model zal soms frequentiepieken zoals medeklinkers als elderspeak aanschouwen en zo een testdata verkeerd labelen. Maar 72% betekent wel dat het model de meeste testdata goed heeft voorspeld! Dus met verdere ontwikkeling kan er wel een AI gemaakt worden die de elderspeak goed kan herkennen. Maar de accuraatheid zal nooit 100% zijn omdat in de praktijk ook geen microfoon zal gebruikt worden van goede kwaliteit en er zullen nog steeds medeklinkers worden uitgesproken en achtergrondgeluid aanwezig zijn.

Conclusie: Elderspeak herkennen aan de hand van de toonhoogte of frequentie is niet de beste manier. Het is niet mogelijk om onderscheid te maken tussen elderspeak en andere pieken die veroorzaakt worden door niet-elderspeak gerelateerde fenomenen. Een AI kan dit beter onderscheiden maar is ook niet feilloos.

De enige manier waarop dit wel zou werken is als we een applicatie ontwikkelen die de gemiddelde toonhoogte van de spreker zelf kan opmeten. Dit met een professionele microfoon die geen achtergrondgeluid opneemt. Bij opname van de persoon gaan we dan

al de medeklinkers uit de spraak gaan filteren. Het is dus mogelijk. Maar zo een applicatie in de praktijk ontwikkelen is niet vanzelfsprekend.

De volledige code hiervan kan u zien in bijlage B5.

```
In [44]: 1 #DataFlair - Calculate the accuracy of our model  
2 accuracy=accuracy_score(y_true=y_test, y_pred=y_pred)  
3 #DataFlair - Print the accuracy  
4 print("Accuracy: {:.2f}%".format(accuracy*100))
```

Accuracy: 72.22%

Figuur 3.1: Accuraatheid voorspelling testdata

4. Conclusie

Uit deze bachelorproef kan men concluderen dat een applicatie ontwikkelen die elderspeak of de nursery tone kan herkennen mogelijk is maar enkel in bepaalde vormen. Dit wil zeggen dat deze applicatie **nooit een perfecte accuraatheid** zal hebben om de nursery tone te onderscheiden of te herkennen. Dit weten we nu omdat een aantal kenmerken van elderspeak niet te herkennen zijn door een applicatie en andere niet consistent juist zullen herkend worden.

Wat wel zou mogelijk zijn is de applicatie te gebruiken als feedback voor de zorgkundige na een gesprek met een patiënt. Dit kan dan een soort van leerschool zijn voor de zorgkundige om zichzelf te verbeteren aan de hand van die feedback. Hierdoor kunnen we de strijd tegen elderspeak winnen of tenminste al verbeteren. Want een live versie van de applicatie die constant een seintje geeft wanneer elderspeak gebruikt wordt is ten eerste moeilijk te ontwikkelen en ten tweede storend en stresserend voor de zorgkundige die aan het werk is.

De verkleinwoorden zijn goed te herkennen en hiermee kan men ook afleiden of er elderspeak wordt gebruikt of niet. Dit is ook mogelijk om te implementeren in een applicatie maar wel zonder machine learning. Het enige wat nog niet perfect is, is dat namen van personen buiten de zorgkundige en de patiënten kunnen herkend worden als verkleinwoorden als ze eindigen op -ke, -je, -kes of -jes. Dus als er gepraat wordt over een kleindochter die Mieke noemt zal dit foutief herkend worden als verkleinwoord. Maar in de feedback ziet men dan ook deze naam staan dus weet men zelf ook wel dat dit geen verkleinwoord is.

De herhalende zinnen kunnen ook herkend worden maar deze wegen minder door voor te beslissen of er elderspeak wordt gebruikt of niet want men kan ook nog steeds een zin **herhalen omdat de patiënt gewoon “Wablieft?” vraagt.** Opnieuw kan de zorgverlener zelf

zien welke zinnen ze heeft herhaald. Hij of zij beslist dan voor zichzelf of die zin terecht was herhaald of niet.

De toonhoogte kan ook goed herkend worden maar enkel met machine learning. De accuraatheid zal ook nooit 100% zijn maar kan wel in de meerderheid van de gevallen correct zeggen of de nursery tone wordt gebruikt. Dit valt ook goed te implementeren in een applicatie.

4.1 Discussie

Het resultaat is een 'Ja' maar er is nog discussie over mogelijk. De applicatie zal niet altijd even precies zijn en sommige kenmerken van elderspeak zullen nooit worden gecontroleerd. Maar dit was wel te verwachten. De onderzoeksvraag was ook moeilijk te beantwoorden met een ja of nee. Het antwoord zit zoals bij veel onderzoeken ergens halfweg. Er zijn ook nog een aantal dingen niet duidelijk:

- Zullen de mensen die deze monitor gebruiken wel degelijk progressie maken?
- Zullen ze de feedback bekijken en dan nog steeds elderspeak gebruiken?
- Zullen mensen die met een zwaar dialect of aan een hoog tempo praten wel verstaan worden door de spraakherkenning?
- Zal de applicatie dezelfde resultaten geven in de praktijk als in een testomgeving? Want er is geen mogelijkheid geweest om de applicatie te testen in een echte werkomgeving.

Er is dus nog genoeg ruimte voor verder onderzoek. De applicatie kan nog getest worden in de praktijk en we kunnen de gevolgen ervan opvolgen. Men kan ook nog de kenmerken die niet onderzocht zijn in dit onderzoek verder onderzoeken en misschien toch een manier vinden om deze te herkennen met spraakherkenning of een frequentiemeter. Dit om de applicatie completer en preciezer te maken.

Dit onderzoek is natuurlijk ook enkel gedaan door 1 persoon. Misschien dat meerdere mensen die meer kennis hebben van Python, spraakherkenning, machine learning en tonen meer kunnen ontwikkelen.

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 wereld neemt het aantal ouderen enorm toe. Dit wordt ook wel de vergrijzing genoemd. Tussen 2015 en 2050 zal de populatie van mensen boven de 60 jaar bijna verdubbelen van 12% naar 22%. (Steverson, 2018) Hierbij komen zorgverleners dus meer en meer in contact met oudere zorgvragers. Het is dus een snel groeiende sector. Ook zorgverleners die in de thuiscontext aan de slag gaan, komen regelmatig in contact met ouderen, nu meer en meer, omdat steeds meer nadruk komt te liggen op het “ageing in place”-beleid. Die term wil zeggen dat de ouderen zo lang mogelijk in hun eigen woning verlangen te verblijven en er oud te worden, eerder dan beroep te doen op institutionele zorg.

Het is belangrijk dat de ouderen sociaal contact blijven houden. Er wordt zelfs gezegd dat sociaal contact een beschermend effect kan hebben op de fysieke en mentale gezondheid van de oudere. Een eerder onderzoek toont aan dat het welzijn van ouderen meer wordt beïnvloed door een netwerk van vrienden dan door familie. (Chopik, 2017) Het hebben van meer dan één type relatie (namelijk vriendschap en familie) heeft een positieve impact op de kwaliteit van het leven van een oudere. Deze vriendschappen bevatten dus ook vriendschappen met zorgverleners.

Hier bevindt zich nu net het probleem. Niet enkel maken zorgverleners niet genoeg tijd vrij om met de ouderen een gesprek te houden of een sociale band te creëren, maar er wordt

ook vaak op een betuttelende of kinderlijke manier tegen hun gecommuniceerd. Deze communicatiestijl, die “elderspeak” wordt genoemd, kan men herkennen aan onder andere een overdreven toonhoogte (“nursery tone”). Verschillende onderzoeken bevestigen dit ook (Mile, 2017) (Simpson, 2014).

En nu tijdens corona verbeterd dat ook niet. Door de angst van corona, gaan mensen zelf meer elderspeak gebruiken. Dat blijkt uit een onderzoek over gedragsveranderingen tijdens een pandemie (Vale, 2020). Mensen weten dat het virus gevaarlijker is voor ouderen, dus gaan ze hen nog meer gaan betuttelen. Daarom is deze bachelorproef nu belangrijker dan ooit.

Met deze bachelorproef ga ik werken aan een applicatie die de communicatie tegenover ouderen gaat analyseren en de “nursery tone” kan onderscheiden van een normale toon. Eerst ga ik natuurlijk onderzoeken in welke mate de gesprekken kunnen geanalyseerd worden en hoe.

A.2 State-of-the-art

Eerst moet ik spraak kunnen omzetten in text. Er zijn een aantal speech-to-tekst technologieën verkrijgbaar en zelf een paar compatibel met programmeertalen die ik ken zodat ik in mijn programma deze technologie kan implementeren. De eerste die me opviel en me werd aangeraden was de google cloud speech API. Deze ziet er gemakkelijk te gebruiken uit maar is helaas niet gratis.

Er zijn ook een paar open-source projecten online voor deze technologie. Hiervan zijn er wel een aantal outdated, en omdat spraakrecognitie altijd geüpdatet en verbeterd wordt zoek ik wel naar een recenter project.

Ik kan werken in de Jupyter Notebook omgeving met Python om dan verschillende audio data te verwerken en er mogelijks machine learning mee uit te oefenen.

A.3 Methodologie

Ook al is deze bachelor proef vooral een proof of concept maken, ga ik toch ook een literatuur studie schrijven. Hierin kan ik onderzoeken waarom deze proef nodig is en hoe de nursery tone te herkennen is. Ook kan ik onderzoeken welke gelijkaardige applicaties of technologieën er al bestaan en ze vergelijken met mijn project.

Eerst ga ik zorgen voor spraakherkenning in mijn applicatie. Wanneer ik spraak kan omzetten in tekst wil ik:

- Verkleinwoorden kunnen herkennen die de woordenschat betuttelend maakt
- De gezamenlijke “we” vorm kunnen herkennen als die te vaak gebruikt wordt
- Simpelere woorden en zinstructuur kunnen herkennen

- Het type vragen herkennen die worden gesteld

Een andere technologie die ik ga moeten gebruiken is een soort toonhoogte (pitch) meter, en een volume (decibel) meter. Hiermee zou ik:

- Het verschil in toonhoogte kunnen meten van een normaal gesprek
- Een zingende manier (op het einde van de zin een hogere toon) van spreken kunnen herkennen
- Luider praten kunnen herkennen

Misschien kan het ook handig zijn om te herkennen wie er praat aan de hand van hun stem. Zodat de applicatie weet of de oudere praat, of de zorgverlener praat of een bezoeker. Maar dit is nog optioneel, ik zie nog wel hoe noodzakelijk dit is.

Het zal ook nodig zijn om een account te maken per gebruiker om de toonhoogte van elke gebruiker op te nemen. Hierdoor kan ik beslissen welke tonen té hoog zijn. Want iedereen praat op een andere toonhoogte. Ook zal ik moeten kunnen onderscheiden wie er aan het praten is. Dit kan ik doen met stem-herkenning technologie. Want ik hoef de patient niet op te nemen, enkel de verpleegkundige.

A.4 Verwachte resultaten

Ik denk dat het mogelijk moet zijn om uiteindelijk de nursery tone te kunnen herkennen en onderscheiden in een dagelijks gesprek. Er zullen waarschijnlijk nog een aantal foutjes inzitten omdat er zelf bij grote technologieën zoals Google Assistant en Siri van Apple nog altijd foutjes zitten. Ik verwacht dus niet dat het programma 100% zal werken zoals het moet. Maar wel dat het werkt voor de meerderheid van de pogingen.

A.5 Verwachte conclusies

De conclusie bij deze resultaten zou dan zijn dat er wel degelijk een verschil is tussen de nursery tone en een normaal gesprek en dat deze kan gemonitord worden in de echte wereld. De conclusie zou dan ook betekenen dat na een tijdje de applicatie te perfectioneren deze misschien in echte ziekenhuizen en rusthuizen zou kunnen gebruikt worden. De applicatie zou dan ook constant moeten geüpdate worden omdat de woordenschat en dialecten constant veranderen. Net zoals bij Google Assistant of Siri.

B. Bijlagen

B.1 Spraakherkenning

Dit is de code die ervoor zorgt dat de ingesproken audio via een microfoon wordt omgezet in digitale tekst door een spraakherkenningssysteem.

```
import speech_recognition as sr

def callback(recognizer, audio):                                # this is called from the
    background thread
    print("Recognizing...")
    # convert speech to text
    try:
        globals.text += recognizer.recognize_google(audio, language="nl-BE") + " "
        print(globals.text)
    except Exception:
        print("Oops! Didn't catch that")

r = sr.Recognizer()
m = sr.Microphone()

with m as source:
    r.adjust_for_ambient_noise(source)                            #gaat calibreren voor
    achtergrond geluid in te schatten
    stop_listening = r.listen_in_background(m, callback) #stop_listening is nu een
    functie die stopt met luisteren
```

Zoals u kan zien wordt er Google gebruikt als spraakherkenningssysteem. De taal die momenteel geselecteerd is, is Nederlands uit België. Dit zou ook kunnen aangepast worden

in een andere taal omdat Google veel talen kan herkennen.

Eerst worden de recognizer en microfoon gedeclareerd als `r` en `m`. Daarna wordt al het achtergrond geluid gekalibreerd zodat de app weet hoeveel decibel de limiet is voor hij begint woorden te herkennen.

Vanaf de functie `stop_listening` wordt gedeclareerd, gaat de app beginnen luisteren en woorden herkennen. Elk herkend woord wordt in `globals.text` gestoken waarmee hier dan verschillende bewerkingen kunnen gedaan worden. Vanaf de functie `stop_listening` gecalld wordt met een `'false'` als parameter gaat de app stoppen met luisteren. Bijvoorbeeld op deze manier:

```
stop_listening(wait_for_stop=False)           #gaat stoppen met luisteren
```

B.2 Verkleinwoorden

Hier ziet u de code waar berekend wordt of een ingesproken woord een verkleinwoord is of niet.

```
with open('bnDims.txt') as f:                  #read alle -ke woorden die
    geen verkleinwoorden zijn
    globals.KE_words = f.read().splitlines()

def calcdims(patientslist):                    #splitst rouwe text in woorden en zet
    verkleinwoorden in 'dims'                 #berekend ook hoeveel verkleinwoorden er zijn in
    'result'

    globals.result = 0
    words = globals.text.split()
    globals.dims = []
    print(words)

    if patientslist != []:
        patients = patientslist[0]
        print(patients)
    else:
        patients = []

    firstNameUser = globals.userFullName.split(' ', 1)[0]
    print(firstNameUser)

    for word in words:
        if word != None:
            endletters_3 = word[-3:]
            endletters_2 = word[-2:]
            if (
                (endletters_2 == 'je' or endletters_2 == 'ke' or endletters_3 ==
                 'kes' or endletters_3 == 'jes')
                and word.lower() not in patients
                and word != firstNameUser
                and len(word) > 3
```

```

        and word.lower() not in globals.KE_words
    ):
        globals.dims.append(word)
        globals.result += 1
if globals.result == 0:
    print('Er zijn geen verkleinwoorden herkend.')
else:
    print('Er zijn', globals.result, 'verkleinwoorden herkend, namelijk:')
    for word in globals.dims:
        print('- ', word)

```

In het begin van de code wordt eerst een bestand met alle woorden die eindigen op -ke maar geen verkleinwoord zijn ingelezen. Dan wordt de ingesproken tekst opgesplitst in verschillende woorden.

De controle of het ingesproken woord een verkleinwoord is loopt als volgt: Het woord moet voldoen aan meer dan 3 letters, het woord moet eindigen op -ke, -je, -kes of -jes, het woord mag niet overeenkomen met de naam van de gebruiker of van de patiënten en het woord mag niet in de ingelezen lijst van woorden voorkomen. Pas als het woord voldoet aan al deze voorwaarden zal het woord toegevoegd worden aan de lijst `globals.dims` en zal deze lijst getoond worden.

B.3 Frequentiemeter

Dit stuk code gaat de frequentie opmeten van audio.

```

# Read in a WAV and find the freq's
import pyaudio
import wave
import numpy as np

chunk = 16384
#32768
#16384

def Average(lst):
    return sum(lst) / len(lst)

# open up a wave
wf = wave.open('audiofiles/triangle.wav', 'rb')
swidth = wf.getsampwidth()
RATE = wf.getframerate()
# use a Blackman window
window = np.blackman(chunk)
# open stream
p = pyaudio.PyAudio()
stream = p.open(format =
                p.get_format_from_width(wf.getsampwidth()),
                channels = wf.getnchannels(),
                rate = RATE,

```

```

        output = True)

# read some data
data = wf.readframes(chunk)
# play stream and find the frequency of each chunk
freqlist = []
while len(data) == chunk*swidth:
    # write data out to the audio stream
    stream.write(data)
    # unpack the data and times by the hamming window
    indata = np.array(wave.struct.unpack("%dh"%(len(data)/swidth),\
                                     data))*window

    # Take the fft and square each value
    fftData=abs(np.fft.rfft(indata))**2
    # find the maximum
    which = fftData[1:].argmax() + 1
    # use quadratic interpolation around the max
    if which != len(fftData)-1:
        y0,y1,y2 = np.log(fftData[which-1:which+2:])
        x1 = (y2 - y0) * .5 / (2 * y1 - y2 - y0)
        # find the frequency and output it
        thefreq = (which+x1)*RATE/chunk
        print ("The freq is %.0f Hz." % (thefreq))
        freqlist.append(thefreq)
    else:
        thefreq = which*RATE/chunk
        print ("The freq is %.0f Hz." % (thefreq))
        freqlist.append(thefreq)
    # read some more data
    data = wf.readframes(chunk)
if data:
    stream.write(data)
freqlistavg = Average(freqlist)
print("Average: %.0.2f Hz." % (freqlistavg))
stream.close()
p.terminate()

```

De variabele 'chunk' die op 16384 staat gedeclareerd is de lengte in bits van een stukje audio waar de frequentie van wordt gemeten. Deze code gaat dus niet 1 frequentie berekenen voor de gehele audio maar de audio splitsen in verschillende kleine deeltjes van de grote die in 'chunk' staat. Voor elk van deze chunks gaat hij dus de frequentie meten.

Er wordt ook nog een gemiddelde berekend van alle frequenties van alle chunks. Dit gemiddelde wordt dan op het einde weergegeven.

B.4 Herhalende zinnen

Deze functie gaat de herhalende zinnen gaan berekenen.

```

def calcrepeaters():
    globals.cache = []

```

```

globals.resultequals = 0
globals.equalwords = []
words = globals.text.split()
equalcounter = 0
toBeDeleted = []

for word in words:
    if word != None:
        while len(globals.cache) >= 25:
            globals.cache.pop(0)
            globals.cache.append(word)

globals.sameequals = {word.lower():globals.cache.count(word) for word in
    globals.cache} #dit lijstje gaat voor elk woord gaan tellen hoeveel keer
    die voorkomt in de cache en hem in een dict steken

for word in globals.sameequals:                                     #add
    single words to delete list
    if globals.sameequals[word] == 1:
        toBeDeleted.append(word)

for word in toBeDeleted:                                           #delete
    words in delete list from dict
    del globals.sameequals[word]

equalcounter = len(globals.sameequals)

globals.resultequals += equalcounter

```

De ingesproken tekst zal opnieuw in woorden opgesplitst worden en daarna in een cache gestoken worden van maximum 25 woorden. Als deze 25 woorden overschreden worden zullen de woorden in het begin van de lijst verwijderd worden zodat er nieuwe woorden achteraan de lijst kunnen toegevoegd worden.

Er wordt dan per woord geteld hoe vaak ze voorkomen en alle woorden die maar 1 keer voorkomen worden verwijderd uit de lijst.

B.5 Emotie-herkenner

Hier ziet u de code van de emotie-herkenner die is omgevormd naar de elderspeak-herkenner.

```

import librosa
import soundfile
import os, glob, pickle
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score

```

```

from pydub import AudioSegment

# Extract features (mfcc, chroma, mel) from a sound file
def extract_feature(file_name, mfcc, chroma, mel):
    with soundfile.SoundFile(file_name) as sound_file:
        X = sound_file.read(dtype="float32")
        sample_rate=sound_file.samplerate
        if chroma:
            stft=np.abs(librosa.stft(X))
        result=np.array([])
        if mfcc:
            mfccs=np.mean(librosa.feature.mfcc(y=X, sr=sample_rate, n_mfcc=40).T,
                           axis=0)
            result=np.hstack((result, mfccs))
        if chroma:
            chroma=np.mean(librosa.feature.chroma_stft(S=stft,
                                                       sr=sample_rate).T,axis=0)
            result=np.hstack((result, chroma))
        if mel:
            mel=np.mean(librosa.feature.melspectrogram(X,
                                                       sr=sample_rate).T,axis=0)
            result=np.hstack((result, mel))
    return result

# Emotions in the RAVDESS dataset
emotions={
    '01':'neutral',
    '02':'elderspeak'
}

# Emotions to observe
observed_emotions=['neutral', 'elderspeak']

# Load the data and extract features for each sound file
def load_data(test_size=0.2):
    x,y=[],[]
    for file in glob.glob("D:\\EigenAudio\\*\\*\\.wav"):
        file_name=os.path.basename(file)
        emotion=emotions[file_name.split("-")[2]]
        if emotion not in observed_emotions:
            continue

        #sound = AudioSegment.from_wav("D:\\EigenAudio\\Joline\\" + file_name)
        #sound = sound.set_channels(1)
        #sound.export("D:\\EigenAudio\\Joline\\" + file_name, format="wav")

        feature=extract_feature(file, mfcc=True, chroma=True, mel=True)
        x.append(feature)
        y.append(emotion)
    return train_test_split(np.array(x), y, test_size=test_size, random_state=9)

# Split the dataset
x_train,x_test,y_train,y_test=load_data(test_size=0.25)

# Get the shape of the training and testing datasets
print((x_train.shape[0], x_test.shape[0]))

```

```
# Get the number of features extracted
print(f'Features extracted: {x_train.shape[1]}')

# Initialize the Multi Layer Perceptron Classifier
model=MLPClassifier(alpha=0.01, batch_size='auto', epsilon=1e-08,
                    hidden_layer_sizes=(300,), learning_rate='adaptive', max_iter=500)

# Train the model
model.fit(x_train,y_train)

# Predict for the test set
y_pred=model.predict(x_test)

# Calculate the accuracy of our model
accuracy=accuracy_score(y_true=y_test, y_pred=y_pred)
# Print the accuracy
print("Accuracy: {:.2f}%".format(accuracy*100))
```

In de functie 'extract_feature' ziet u de 3 gebieden waar de audio op gecontroleerd wordt. U ziet dat deze alle 3 uit de librosa library komen.

De functie 'load_data' gaat de ingesproken audio inladen. De code in commentaar in deze functie moet maar 1 keer gecompileerd worden voor elke spreker. Deze code zorgt er voor dat de audio monotoon wordt in plaats van stereotoon. Dit moet omdat de machine learning feature enkel met monotone audio kan werken.

Dan ziet u dat de train en test data worden opgevuld met de audio, het model wordt geïnitieerd en dan getraind. Op het einde laten we het model een voorspelling doen op testdata en wordt de accuraatheid getoond.

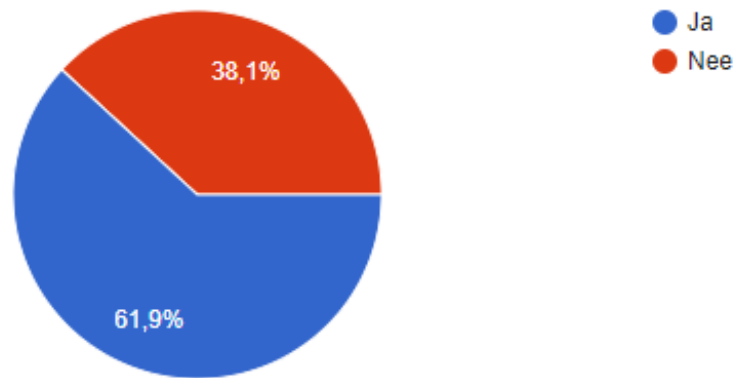
B.6 Enquête

Hier kan u alle resultaten zien van de enquête die door 97 mensen is ingevuld.

De vraag met in het rood 'Student' boven is enkel voor de studenten. De vraag met in het rood 'Werkende mensen' boven is enkel voor de niet-studenten. Bij de vragen met de bar grafieken staat 1 voor oneens en 5 voor eens.

Bent u student?

97 antwoorden

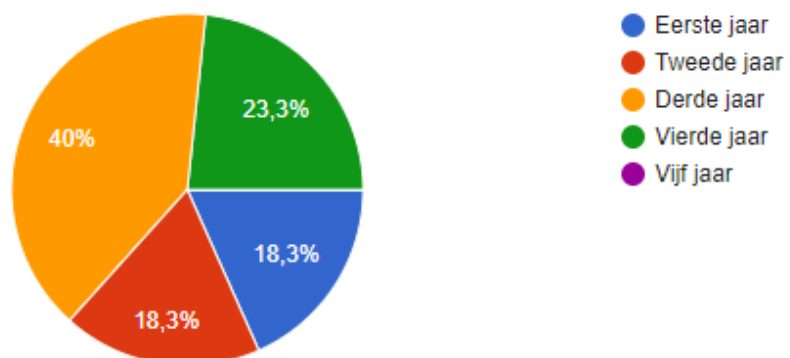


Figuur B.1: Resultaat vraag 1
Source: Victor Standaert

Student

In welk jaar van uw opleiding zit u?

60 antwoorden

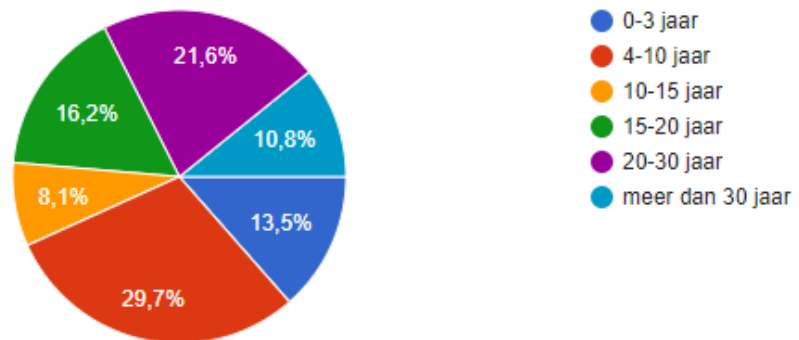


Figuur B.2: Resultaat vraag 2
Source: Victor Standaert

Werkende mensen

Hoe lang werkt u al binnen de zorg- / verpleegkunde?

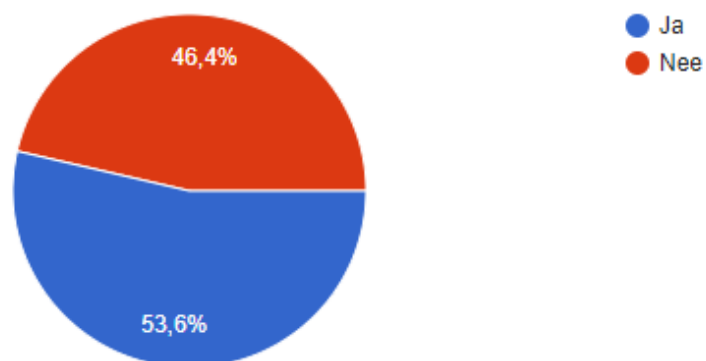
37 antwoorden



Figuur B.3: Resultaat vraag 3
Source: Victor Standaert

Had u voor deze vragenlijst al gehoord van elderspeak of de nursery tone?

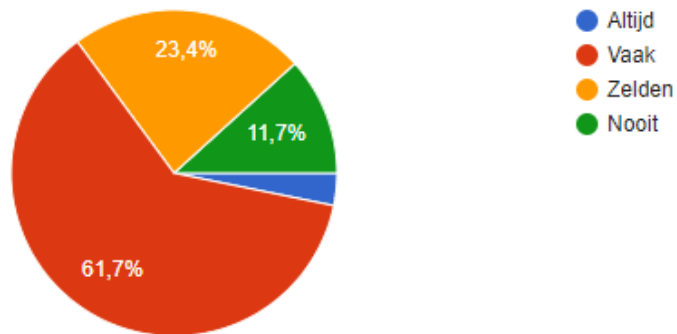
97 antwoorden



Figuur B.4: Resultaat vraag 4
Source: Victor Standaert

Komt u op uw stage of werk vaak in contact met elderspeak? (bij uzelf of anderen)

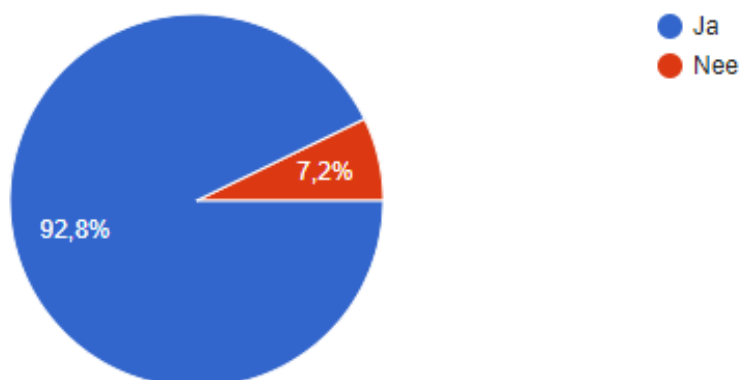
94 antwoorden



Figuur B.5: Resultaat vraag 5
Source: Victor Standaert

Denkt u dat elderspeak een probleem is dat moet aangepakt worden?

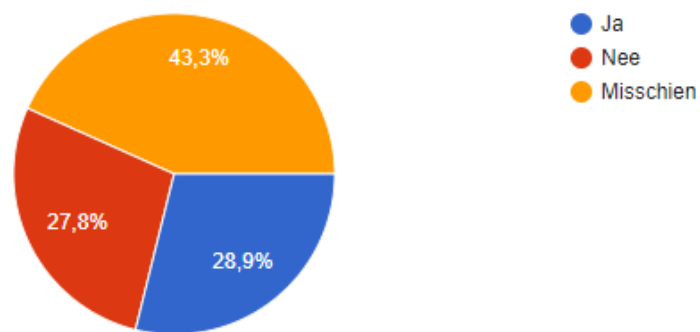
97 antwoorden



Figuur B.6: Resultaat vraag 6
Source: Victor Standaert

Zou u een monitor op zak willen die een seintje geeft telkens u elderspeak gebruikt?

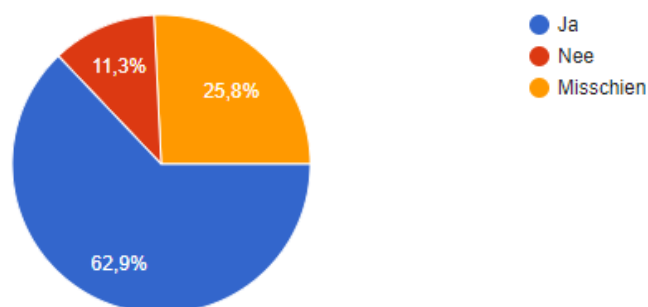
97 antwoorden



Figuur B.7: Resultaat vraag 7
Source: Victor Standaert

Zou u na een gesprek met een oudere een overzicht willen met daarop feedback over hoe vaak u de nursery tone heeft gebruikt tijdens dat gesprek?

97 antwoorden

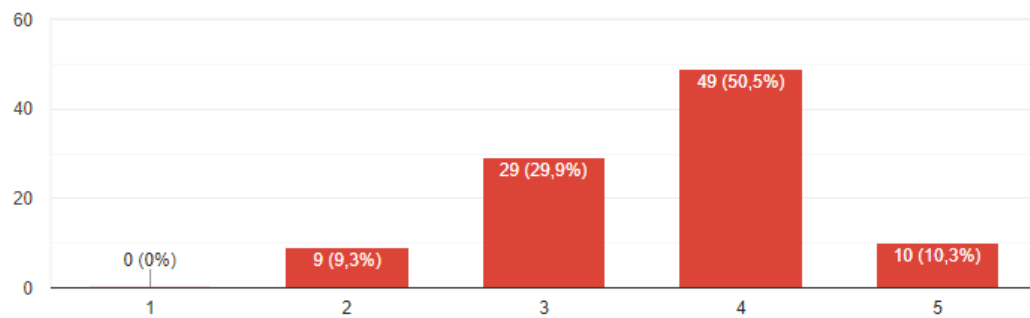


Figuur B.8: Resultaat vraag 8
Source: Victor Standaert

Een nursery tone monitor zou de strijd tegen elderspeak kunnen helpen.



97 antwoorden

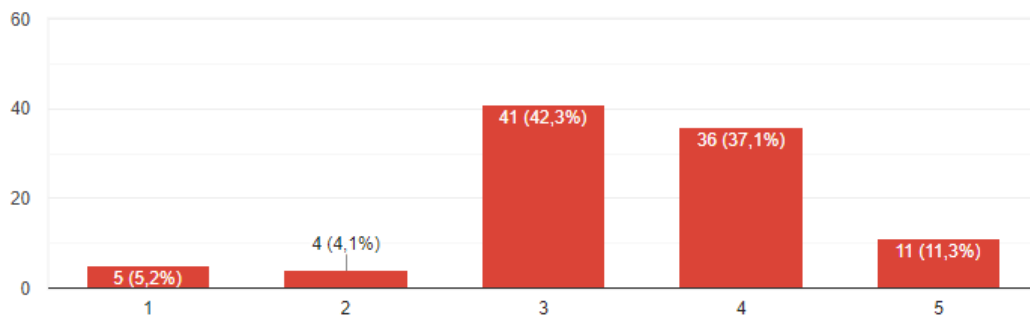


Figuur B.9: Resultaat vraag 9
Source: Victor Standaert

Ik zou met een nursery tone monitor minder elderspeak gebruiken.



97 antwoorden

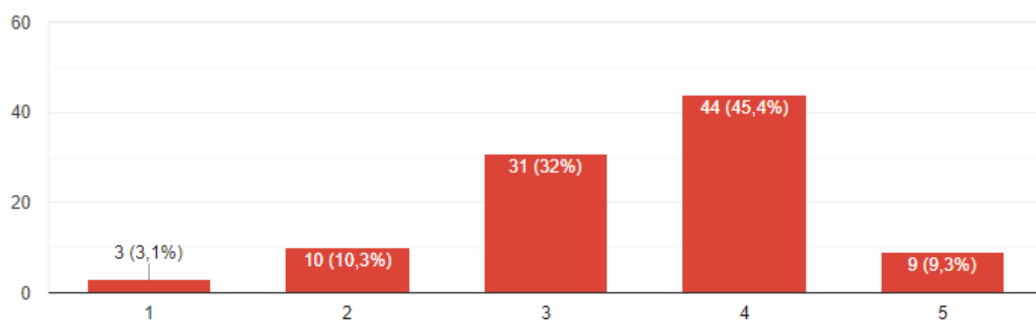


Figuur B.10: Resultaat vraag 10
Source: Victor Standaert

Een nursery tone monitor zou meer stressgevend zijn voor de verpleeg- of zorgkundigen.



97 antwoorden



Figuur B.11: Resultaat vraag 11
Source: Victor Standaert

Bibliografie

- Abrahms, S. (2017). *Identifying and combating elderspeak*. <https://extramile.thehartford.com/lifestyle/aging/elderspeak/>
- Bohouta, K., Gamal. (2017). *Comparing Speech Recognition Systems (Microsoft API, Google API And CMU Sphinx)*. https://www.researchgate.net/publication/314938892_Comparing_Speech_Recognition_Systems_Microsoft_API_Google_API_And_CMU_Sphinx
- Chopik, W. (2017). *Associations among relational values, support, health, and well-being across the adult lifespan*. <https://onlinelibrary.wiley.com/doi/abs/10.1111/pere.12187>
- Discrete Fourier Transform (numpy.fft)*. (2021). <https://numpy.org/doc/stable/reference/routines.fft.html>
- Duffy, M. (2020). *What's So Wrong with "Elderspeak," Anyway? Answer: Everything*. <https://visionaware.org/blog/visionaware-blog/whats-so-wrong-with-elderspeak-anyway-answer-everything/>
- Encyclo.nl. (2021). *3919 woorden eindigen op ke*. <https://www.encyclo.nl/woorden-die-eindigen-op-ke/1&xls=1>
- Gouveia, M., Odília Maria Rocha. (2016). *Social networks and quality of life of elderly persons: a review and critical analysis of literature*. <https://doi.org/10.1590/1981-22562016019.160017>
- Identifying speakers with voice recognition*. (g.d.). https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781787125193/9/ch09lvl1sec61/identifying-speakers-with-voice-recognition
- Kaspar, D. (2019). *Librosa: A Python Audio Library*. <https://medium.com/@patrickbfuller/librosa-a-python-audio-library-60014eeacfb>
- Kinsella, B. (2018). *Google Assistant Takes Lead in Understanding Speakers with Accents*. <https://voicebot.ai/2018/09/13/google-assistant-takes-lead-in-understanding->

- speakers - with - accents/#: %7E: text=Google % 20Continues % 20to % 20Lead % 20in % 20Speech % 20Recognition& text=Google % 20again % 20stood % 20out % 20as, is % 20the % 20variance % 20across % 20accents.
- Lenke, N. (2017). *Does speech recognition understand your dialect?* <https://www.itproportal.com/features/does-speech-recognition-understand-your-dialect/>
- Livingstone, S. (2018). *The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS)*. <https://zenodo.org/record/1188976>
- Microphones, D. (2021). *FACTS ABOUT SPEECH INTELLIGIBILITY*. <https://www.dpamicrophones.com/mic-university/facts-about-speech-intelligibility>
- Mile, E. (2017). *Identifying and Combating Elderspeak*. <https://extramile.thehartford.com/lifestyle/aging/elderspeak/>
- Pfitzinger, J. (2019). *The Negative Effects of Elderspeak*. <https://www.nextavenue.org/negative-effects-elderspeak/>
- Python Mini Project – Speech Emotion Recognition with librosa*. (2021). <https://data-flair.training/blogs/python-mini-project-speech-emotion-recognition/>
- Simpson, J. (2014). *Elderspeak – Is it helpful or just baby talk?* <https://merrill.ku.edu/elderspeak-it-helpful-or-just-baby-talk>
- Steverson, M. (2018). *Ageing and health*. <https://www.who.int/news-room/fact-sheets/detail/ageing-and-health>
- Thomas, M. (2020). *Neural Networks: Advantages and Applications*. <https://www.marktechpost.com/2019/04/18/introduction-to-neural-networks-advantages-and-applications/#:~:text=Advantages%20of%20Neural%20Networks%3A,does%20not%20affect%20its%20working.>
- Vale, M. (2020). *Ageism and Behavior Change During a Health Pandemic: A Preregistered Study*. <https://www.frontiersin.org/articles/10.3389/fpsyg.2020.587911/full>
- Verkleinwoorden - hoofdregel | woordenlijst*. (g.d.). <https://woordenlijst.org/leidraad/15/1>
- Wikipedia. (2019). *Elderspeak*. <https://en.wikipedia.org/wiki/Elderspeak>