

Elderspeak Detector: tekstanalyse met behulp van AI voor het detecteren van ‘Secondary Babytalk’ in de zorgverlening.

Ine Van den Branden.

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

Promotor: Mevr. S. De Vreese

Co-promotor: Dhr. J. Campens

Academiejaar: 2023–2024

Eerste examenperiode

Departement IT en Digitale Innovatie .

**HO
GENT**

Woord vooraf

Voor mijn bachelorproef wou ik graag iets doen wat uiteindelijk een verschil zou kunnen maken in de maatschappij. Er mee voor kunnen zorgen dat ouderen op een correcte en respectvolle manier kunnen behandeld worden past dus zeker in deze visie. Daarnaast heb ik ook een grote interesse voor taal en hoe computers kunnen ingezet worden om taal te begrijpen en analyseren. Vandaar dat ik tijdens mijn Erasmus-uitwisseling een vak heb gevolgd over tekstanalyse en *language processing*, wat zeker mijn interesse voor dit vakgebied heeft vergroot, en een goede kennisbasis vormde om met dit bachelorproefonderwerp aan de slag te kunnen gaan.

Tot slot was dit onderwerp ook een kans om met de nieuwste AI-technologieën te werken. Artificiële intelligentie is een snel evoluerend vakgebied, wat het boeiend maakt om er steeds iets nieuws over te weten te komen en zich er verder in te verdiepen.

Zelfs tijdens het schrijven van deze bachelorproef werden er voortdurend nieuwigheden en verbeteringen bekend gemaakt door grote spelers binnen de AI-wereld, zoals OpenAI en Google. Twee jaar geleden bestonden sommige technologieën die beschreven staan in deze bachelorproef zelfs nog niet.

Graag wil ik ook alle personen bedanken die me geholpen en gesteund hebben tijdens het schrijven van deze scriptie.

Ten eerste bedank ik mijn promotor, Sabine De Vreese, en mijn co-promotor, Jorrit Campens voor de feedback en tips, en om wekelijks tijd vrij te maken om samen te zitten over de voortgang van de bachelorproef.

Daarnaast bedank ik ook mijn familie en vrienden, voor de steun doorheen deze periode. Er waren zeker moeilijke momenten waarbij hun steun heel erg veel betekende voor mij.

Ine Van den Branden

Samenvatting

Dit onderzoek heeft als hoofddoel om het communiceren met ouderen in zorginstellingen te verbeteren aan de hand van technologie. Door betuttelend taalgebruik, zoals *Elderspeak*, automatisch met een computerprogramma te detecteren kunnen zorgverleners erop gewezen worden wat ze precies fout doen. Zo kunnen ze zelf aan de hand van deze informatie in de toekomst op een betere en respectvollere manier met patiënten omgaan. Omdat er steeds meer ouderen in rusthuizen terechtkomen kan een dergelijke oplossing een meerwaarde bieden aan onze maatschappij.

Het doel van dit eindwerk was om de bestaande *Elderspeak*-applicatie te verbeteren, waar reeds in vorige jaren binnen HOGENT onderzoek naar werd uitgevoerd. Deze applicatie zou idealiter worden ingezet binnen de opleiding verpleegkunde aan de HOGENT om studenten aan te leren wat *Elderspeak* is en hoe ze het gebruik ervan kunnen vermijden.

Één van de mogelijke verbeterpunten was om de bestaande algoritmen die ervoor instaan om de tekstuele kenmerken van *Elderspeak* uit een tekst te extraheren, te verbeteren aan de hand van Artificiële Intelligentie (AI). De onderzoeksvraag van deze bachelorproef luidt als volgt: “Hoe kan Artificiële Intelligentie worden ingezet om *Elderspeak* te detecteren op basis van tekstuele kenmerken?”

Er werd onderzoek gedaan of een *Large Language Model* zoals GPT-3.5 Turbo, wat bekend is vanwege het gebruik ervan in de chatbot ChatGPT, kan ingezet worden om *Elderspeak* te herkennen in een tekst. Daaruit bleek dat het model wel begreep wat verstaan wordt onder *Elderspeak*, maar het lastig had om zelf tekstuele kenmerken zoals verkleinwoorden aan te duiden.

Een betere oplossing bleek het gebruik van AI die gespecialiseerd is in het uitvoeren van tekstanalyse, met name de *Part-of-Speech Tagger* van spaCy. De accuraatheid van het detecteren van verkleinwoorden, tussenwerpsels en collectieve voornaamwoorden ten opzichte van het reeds bestaande programma werd hiermee verbeterd van 92,71% naar 99,48%.

Inhoudsopgave

Lijst van figuren	vii
--------------------------	------------

Lijst van tabellen	viii
---------------------------	-------------

1 Inleiding	1
1.1 Probleemstelling	1
1.1.1 Age-isme	1
1.1.2 Vergrijzing	2
1.1.3 Secondary babytalk	3
1.2 Onderzoeksvraag	3
1.3 Onderzoeksdoelstelling	4
1.4 Opzet van deze bachelorproef	4
2 Stand van zaken	5
2.1 Wat is taal?	5
2.2 Taal en computers	5
2.3 Natural Language Processing	6
2.3.1 Syntactische Analyse	6
2.3.2 Semantische Analyse	7
2.3.3 Generatieve NLP	7
2.4 AI en Machine Learning	7
2.4.1 Neurale netwerken	7
2.4.2 Transformers	8
2.5 Large Language Models	10
2.5.1 Propriëtaire LLM's	11
2.5.2 Open Source LLM's	13
2.5.3 Vergelijking kwaliteit, snelheid en prijs	13
3 Methodologie	14
3.1 Verzameling van de testdata	14
3.1.1 Opnames	14
3.1.2 Verwerking	15
3.1.3 Transcripties	15
3.2 Classifiers	15
3.3 Large Language Models	16
3.4 Verbetering tekstanalyse	16

4 Resultaten	17
4.1 Resultaten GPT-3.5	17
4.1.1 Genereren van gesprekken	17
4.1.2 Detecteren van <i>Elderspeak</i> in zinnen	18
4.1.3 Detecteren van verkleinwoorden	19
4.1.4 Detecteren van woordsoorten in transcripties	19
4.2 Part of Speech Tagging	20
4.2.1 Onderzochte woordsoorten	21
4.2.2 Resultaat en vergelijking	22
4.3 Verbetering Lay-out	24
4.3.1 Beschrijving functionaliteit	24
4.3.2 Aanpassingen	26
5 Vervolgonderzoek	28
5.1 Speaker Diarization	28
5.2 Alternatief model	28
5.2.1 Assembly AI	29
5.3 Resultaten	29
6 Conclusie	31
6.1 Bijdrage van dit onderzoek	31
6.2 Gebruik van LLM's	31
A Onderzoeksvoorstel	33
A.1 Introductie	33
A.2 State-of-the-art	34
A.2.1 Natural Language Processing	34
A.2.2 Classifiers	34
A.2.3 Voorgaand onderzoek	34
A.3 Methodologie	35
A.4 Verwacht resultaat, conclusie	36
B Resultaten	37
B.1 Repositories	37
B.2 Resultaten GPT-3.5	37
B.3 Resultaten spaCy	37
Bibliografie	45

Lijst van figuren

1.1	Gemiddelde leeftijd Belgische bevolking	2
1.2	Aantal 80-plussers in een collectief huishouden	3
2.1	Deep Learning	8
2.2	Transformer-architectuur	9
2.3	Visualisatie Attentie-mechanisme	10
4.1	Accuraatheid oud programma	23
4.2	Accuraatheid nieuw programma	24
4.3	Oude <i>Elderspeak</i> -detectie applicatie	25
4.4	Nieuwe <i>Elderspeak</i> -detectie applicatie	27

Lijst van tabellen

5.1	Accuraatheid Speech-to-Text modellen	30
-----	--	----

1

Inleiding

1.1. Probleemstelling

Vanwege de vergrijzende bevolking komen er meer en meer ouderen in rusthuizen terecht. Daarom is het belangrijk om deze bevolkingsgroep juiste en passende ondersteuning te bieden. Opleidingen verpleegkunde zetten reeds in in het opleiden van verpleegkundigen met de juiste kennis en vaardigheden. Een belangrijke vaardigheid is het vermogen om op een juiste en respectvolle manier met de patiënten te communiceren.

Er bestaan bepaalde vooroordelen tegenover de ouderen vanwege hun leeftijd. Dit staat bekend als age-isme. *Elderspeak* is de term voor het betuttelen van oudere bewoners van rusthuizen en andere zorginstellingen. Zij kunnen er negatieve gevolgen van ondervinden.

Hieronder worden deze termen verder besproken en welke invloed ze kunnen hebben op het mentale welzijn van rusthuisbewoners.

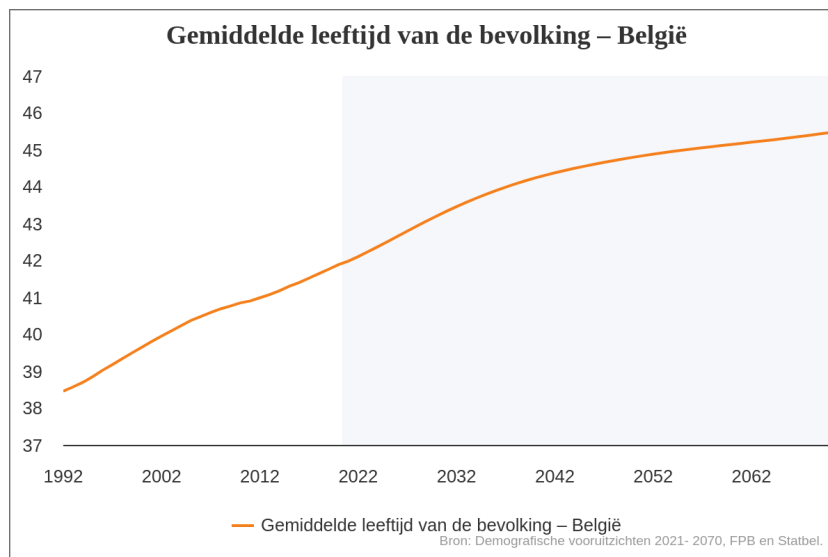
1.1.1. Age-isme

Smit en van Vuuren (2023) beschrijven age-isme als volgt:

“Age-isme is een verzamelnaam voor stereotypering, vooroordelen en discriminatie op basis van leeftijd.”

Volgens het onderzoek van Smit en van Vuuren (2023) zijn er drie verschillende vormen van age-isme: interpersoonlijk age-isme, op jezelf gericht age-isme en institutioneel age-isme.

- Interpersoonlijk age-isme gaat over discriminatie die voorkomt tussen verschillende personen.
- Bij op jezelf gericht age-isme gaat de persoon zichzelf op een discriminerende manier beoordelen. Hierbij bestaan er twee vormen: ten eerste kan de per-



Figuur (1.1)

Gemiddelde leeftijd Belgische bevolking. Bron: Duyck e.a. (2022)

soon denken dat hun mogelijkheden gelimiteerd zijn vanwege hun leeftijd. Ten tweede is er ook de angst om gediscrimineerd en gestereotypeerd te worden. Dit fenomeen is ook bekend als *stereotype threat*. Beide vormen kunnen een soort *self-fulfilling prophecy* vormen omdat de persoon onzekerder wordt en zo mogelijk de stereotypes kan bevestigen.

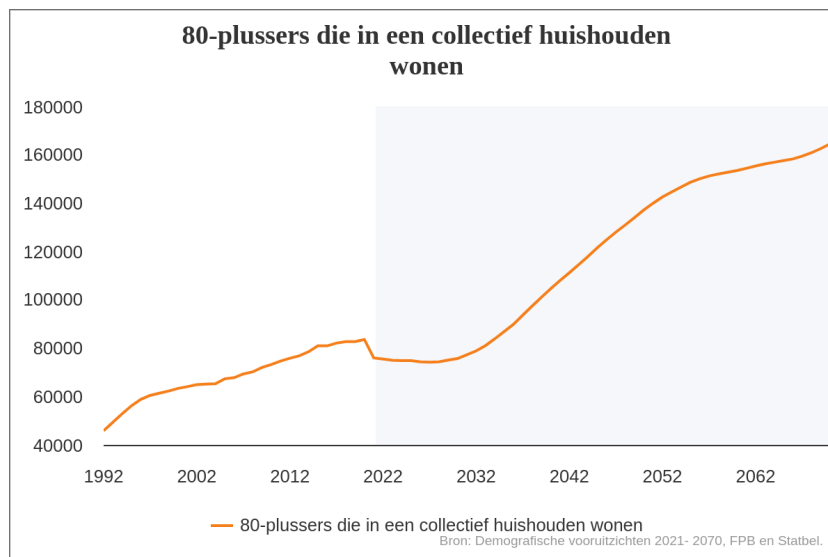
- Institutioneel age-isme gaat eerder over discriminerende wetten en regels die bepaald worden door officiële instituties zoals overheden of bedrijven.

Merton (1967) zegt het volgende over de term *self-fulfilling prophecy*:

“The self-fulfilling prophecy is, in the beginning, a false definition of the situation evoking a new behavior which makes the original false conception come true. This specious validity of the self-fulfilling prophecy perpetuates a reign of error. For the prophet will cite the actual course of events as proof that he was right from the very beginning.”

1.1.2. Vergrijzing

Alle vormen van age-isme creëren een probleem in de maatschappij, omdat het ervoor zorgt dat een hele bevolkingsgroep onjuist wordt behandeld. Zeker nu steeds een groter percentage van de bevolking tot deze groep behoort. Volgens Duyck e.a. (2022), statistieken van het Federaal Planbureau (FPB) en Statbel was de gemiddelde leeftijd in 2022 42 jaar. Die leeftijd stijgt tot 44,8 jaar in 2050 en tot 45,5 jaar in 2070. Dit wordt verduidelijkt in figuur 1.1. Bij het aantal 80-plussers die in een collectief huishouden wonen (Figuur 1.2) is er ook een opwaartse trend zichtbaar.

**Figuur (1.2)**

Aantal 80-plussers in een collectief huishouden. Bron: Duyck e.a. (2022)

1.1.3. Secondary babytalk

Elderspeak

Elderspeak is een vorm van interpersoonlijk age-isme, omdat het gaat over een interactie tussen verschillende personen, waarvan er minstens één op oudere leeftijd is.

“Elderspeak is een vereenvoudigd spraakregister dat wordt gebruikt bij oudere volwassenen en klinkt als babypraat. Het wordt gekenmerkt door een verscheidenheid aan taalkundige aanpassingen in ritme, klank, zinsstructuur en betekenis, zoals een hoge en overmatig koesterende stem, het gebruik van ongepaste uitingen van genegenheid (bijv. ‘lieverd’) en vervangend gebruik van collectieve voornaamwoorden (bijv. , ‘wij’ in plaats van ‘jij’).” (Shaw & Gordon, 2021)

Het gebruik van *Elderspeak* kan als betuttelend overkomen voor ouderen en een slechte invloed hebben op hun mentaal welzijn. Zo kunnen ze een lager zelfbeeld krijgen. Bij personen met dementie kan het ook leiden tot ongewillig gedrag tegenover zorgverleners.

Nursery tone

Nursery tone is de term voor de intonatie die kenmerkend is voor *Elderspeak*. Hierbij is de toonhoogte op het einde van een zin hoger dan normaal, wat een infantiliserende indruk kan geven.

1.2. Onderzoeksvraag

Om het probleem van *Elderspeak* aan te pakken is het belangrijk om het juist en automatisch te kunnen detecteren. Daarbij kunnen verschillende technologieën worden toegepast. Eerst en vooral moet gesproken tekst worden omgezet naar

geschreven tekst. Daarvoor werd reeds in het voorgaande onderzoek rond dit onderwerp een speech-to-text algoritme ingezet. Op deze tekst wordt dan analyse toegepast die bepaalt welke delen van de tekst aansluiten bij kenmerken van *Elderspeak*, en deze aanduidt. Er waren een aantal verbeterpunten aanwezig in de bestaande applicatie, waar dit onderzoek zich vooral op toespitst. De onderzoeksvraag voor dit onderzoek is: “Hoe kan Artificiële Intelligentie worden ingezet om *Elderspeak* te detecteren op basis van tekstuele kenmerken?” Daarnaast ook: “Hoe kan de bestaande applicatie worden verbeterd, op basis van lay-out en bruikbaarheid?”

1.3. Onderzoeksdoelstelling

Het doel van dit onderzoek is om de bestaande Elderspeak-detectie applicatie te verbeteren, dit vooral op basis van het detecteren van tekstuele kenmerken. Het onderzoek is geslaagd wanneer de tekstanalyse beter presteert dan wat er in voorgaand onderzoek werd ontwikkeld. De lay-out van de applicatie wordt verbeterd zodat het meer aansluit bij de wensen van de gebruiker en de gebruikssituaties waar het in de praktijk voor wordt ingezet, namelijk het analyseren van videofragmenten.

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 vervolgens de verkregen resultaten opgelijst en besproken. Hoofdstuk 5 beschrijft kort wat de mogelijke manieren zijn om de applicatie te verbeteren.

In Hoofdstuk 6, 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

Dit hoofdstuk beschrijft wat de huidige stand van zaken is binnen het vakgebied. Eerst wordt kort een definitie gegeven van het begrip *taal*. Vervolgens wordt een overzicht gegeven over Natural Language Processing (NLP) en de verschillende toepassingen ervan. Daarna volgt een beschrijving van AI en machine learning-technieken die relevant zijn voor dit onderzoek. Tot slot volgt een opsomming en vergelijking van de verschillende AI-modellen die beschikbaar zijn en mogelijk een meerwaarde vormen bij het verbeteren van de *Elderspeak*-applicatie.

2.1. Wat is taal?

“Language, a system of conventional spoken, manual (signed), or written symbols by means of which human beings, as members of a social group and participants in its culture, express themselves. The functions of language include communication, the expression of identity, play, imaginative expression, and emotional release.” (Robins & Crystal, [2024](#))

Het gebruiken van taal is wat mensen onderscheidt van andere dieren. Een van de verschillen is dat menselijke taal ook de toekomst en het verleden kan beschrijven. Daarnaast staat het mensen ook toe om zich uit te drukken over een breed spectrum, waaronder ook emoties. Één van de kenmerken van natuurlijke taal is ambiguïteit, ofwel dezelfde betekenis op verschillende manieren kunnen uitdrukken.

2.2. Taal en computers

Vanwege deze complexiteit is het lastig om een computer met taal te laten werken. Ten eerste begrijpen computers enkel cijfers, geen woorden. Het belangrijkste vraagstuk in dit vakgebied is dus hoe op een efficiënte manier woorden kunnen

omgezet tot een numerieke representatie die kan begrepen worden door een computersysteem.

Daarnaast heeft taal weinig vaste regels, het verandert voortdurend en is flexibel. Sommige woorden hebben een andere betekenis afhankelijk van de context.

Omdat *Elderspeak* vaak over subtiele verschillen gaat, is het detecteren ervan met een computer een grote opgave.

Tot slot verschilt het ook van persoon tot persoon hoe iets wordt geïnterpreteerd, wat iemand als neerbuigend ziet kan voor een andere persoon als vriendelijk worden beschouwd.

2.3. Natural Language Processing

Natural Language Processing of NLP (natuurlijke taalverwerking in het Nederlands) is een onderzoeksgebied dat zich toespitst op het onderzoeken van natuurlijke taal en hoe men er een machine kan mee laten omgaan. Zo kan dit ingezet worden voor verschillende toepassingen zoals vertalen of tekstanalyse.

De toepassingen van NLP kunnen opgedeeld worden in analyse-toepassingen en generatieve toepassingen (Lipizzi e.a., 2019). Deze twee staan ook bekend als Natural Language Understanding (NLU) en Natural Language Generation (NLG). Hieronder worden de belangrijkste toepassingen die nuttig kunnen zijn voor dit onderzoek opgelijst.

2.3.1. Syntactische Analyse

Bij syntactische analyse wordt de grammaticale structuur van tekst onderzocht.

Tokenisatie

Tokenisatie in de context van NLP is het proces waarbij tekst wordt omgezet in kleinere delen zoals woorden of tekens.

Lemmatisatie

Bij lemmatisatie wordt de basisvorm van een woord geëxtraheerd. Bijvoorbeeld bij werkwoorden is dit de 'onvervoegde' vorm of infinitief van het werkwoord: 'bleef' wordt dan gelemmatiseerd tot 'blijven'. Ook andere woorden hebben een basisvorm, zoals verkleinwoorden, meervouden, en andere vervoegingen. Het correct lemmatiseren van een woord is ook afhankelijk van de positie van het woord in de zin. Hierbij wordt ook gebruik gemaakt van een woordenboek.

Stemming

Stemming is grotendeels gelijk aan lemmatisatie maar houdt geen rekening met de context van het woord.

Part of Speech Tagging (POS)

POS-tagging is een NLP-techniek met als doel 'parts of speech' of woordsoorten toe te kennen aan woorden of *tokens*. Voorbeelden van woordsoorten zijn werk-

woord, bijvoegsel of zelfstandig naamwoord. Omdat de woordsoort van een woord in een zin afhankelijk kan zijn van de context, is het ook de taak van een *tagger* om hierin te kunnen onderscheiden.

2.3.2. Semantische Analyse

Semantische analyse probeert de betekenis van woorden en tekst te achterhalen aan de hand van het verband tussen woorden en de context.

Tekstclassificatie

Bij tekstclassificatie wordt een gebruikergedefinieerd label toegekend op basis van de inhoud van tekst. Dit kan bijvoorbeeld gaan over het detecteren van spam e-mails, of het sorteren van documenten per onderwerp.

Sentimentanalyse

Sentimentanalyse is een vorm van tekstclassificatie dat het sentiment van tekst kan bepalen. Dit is nuttig om bijvoorbeeld na te gaan wat de gemiddelde mening is op sociale media over een bepaald onderwerp. Het kan bepalen of berichten een positieve of negatieve boodschap hebben.

2.3.3. Generatieve NLP

Tekstgeneratie

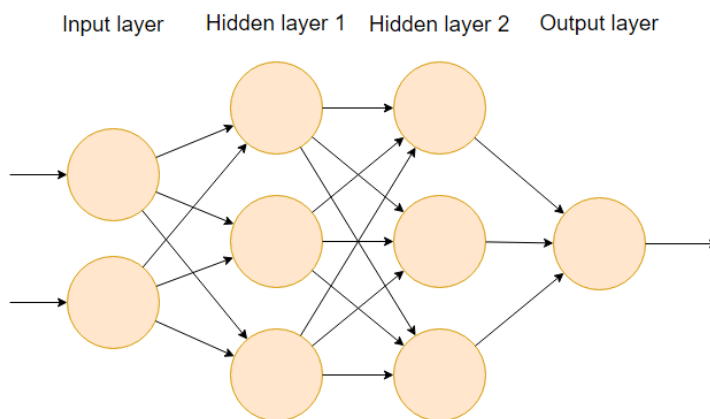
Tekstgeneratie is een NLP-toepassing waarbij een AI-model telkens het volgende woord of token voorspelt en zo teksten kan genereren. Deze modellen leren patronen te herkennen in de grote hoeveelheid aan tekst waar ze mee getraind worden. Na het trainen kunnen ze deze dan reproduceren en zo natuurlijke taal imiteren.

2.4. AI en Machine Learning

Artificiële Intelligentie of AI stelt computers in staat om menselijke intelligentie na te bootsen. AI is vandaag overal aanwezig en vrijwel iedereen heeft er al van gebruik gemaakt. Denk maar aan toepassingen zoals Google Maps of Google Translate. Deze zijn gebaseerd op AI algoritmes. Het is een snel evoluerende technologie, en de toepassingen die vandaag mogelijk zijn, waren enkele jaren geleden niet verwacht.

2.4.1. Neurale netwerken

Neurale netwerken trachten het menselijke brein na te bootsen door een netwerk van 'neuronen' te gebruiken. In elk neuron wordt er informatie opgeslagen zodat wanneer een netwerk van zulke neuronen wordt gebouwd, het model kan 'nadenken' en van een bepaalde input naar een output kan gaan die gerelateerd is aan de input, zelfs als het model deze specifieke input nog nooit eerder heeft gezien. Wanneer er verschillende lagen worden gebruikt is er sprake van *deep learning*.

**Figuur (2.1)**

Een deep learning neurale netwerk

2.4.2. Transformers

Het Transformer-model is een vorm van deep learning die werd geïntroduceerd in de paper getiteld 'Attention is All you Need' in 2017 door onderzoekers van Google en de universiteit van Toronto (Vaswani e.a., [2017](#)). Hieronder worden de verschillende onderdelen van een Transformer-model verder toegelicht.

Embeddings

Computers kunnen niet zomaar met tekst werken, ze begrijpen enkel cijfers. Embeddings zijn numerieke representaties van data die gebruikt worden in machine learning. Ze worden voorgesteld als vectoren. Gerelateerde data zal een vector-representatie toegekend krijgen die dichter bij elkaar ligt dan data die ongerelateerd is.

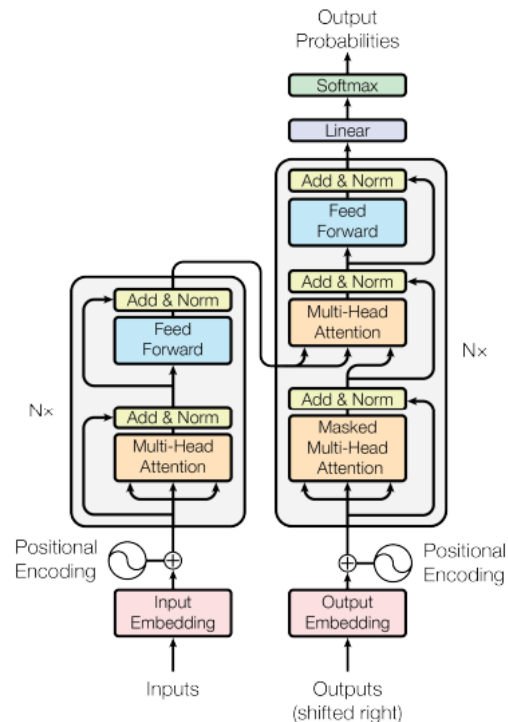
Positional Encoding

Positional encoding zorgt ervoor dat aan elke embedding een relatieve positie wordt toegekend. Daardoor kan elk token een unieke representatie toegekend krijgen door de lagen die volgen, op basis van de context.

Transformer Encoders

Encoders zetten data om in embedding waarin ook de context van de data zit opgeslagen. Een encoder in een Transformer-model bestaat uit verschillende encoder-blokken. De encoder-blokken zelf bestaan uit verschillende lagen, waaronder self-attention en feed-forward. Door middel van self-attention, in transformers ook wel multi-head-attention genoemd, kan de encoder bepalen welke delen van de input belangrijk zijn en hoe ze de context beïnvloeden. Dankzij het feed-forward-netwerk kan het model een diepere betekenis uit de inputgegevens halen en de input compacter en op een nuttigere manier weergeven. (Malingan, [2023](#))

Na beide lagen wordt normalisatie toegepast, wat erin resulteert dat de tijd die nodig is om het model te trainen gereduceerd wordt. Dit gebeurt door ervoor te

**Figuur (2.2)**

Architectuur van het Transformer-model. Bron: Vaswani e.a. (2017)

zorgen dat de inputs een consistente distributie hebben. (Ba e.a., 2016)

Encoders kunnen gebruikt worden voor bijvoorbeeld tekstclassificatie of computervertaling.

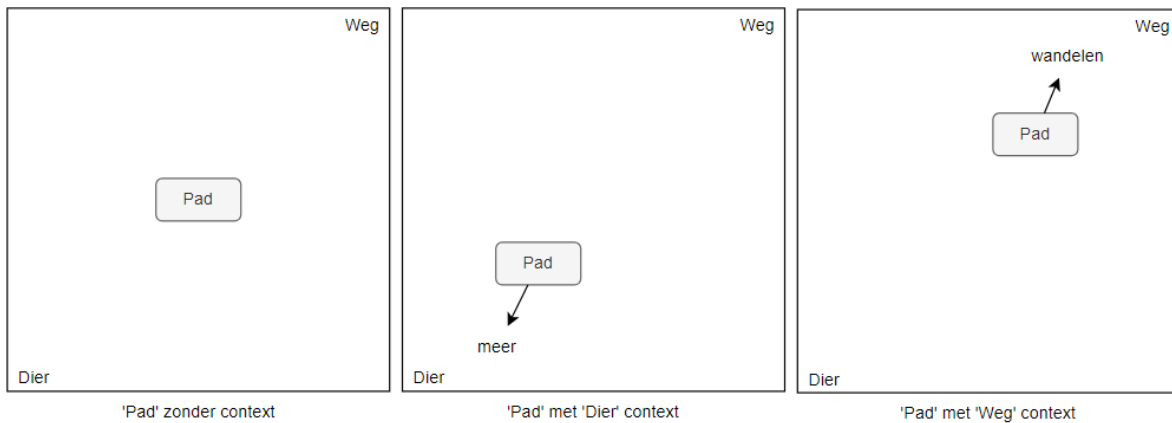
Transformer Decoders

Decoders nemen een input en zetten deze om naar een nieuwe output naargelang waarvoor het model getraind is. Bijvoorbeeld in het geval van tekst-generatie is de input tekst, en bepaalt de decoder het volgende woord of token als output. Deze output wordt daarna het laatste deel van de volgende input. Dit proces heet auto-regressie.

Net zoals encoders bestaan decoders uit verschillende decoder-blokken, waarin telkens een self-attention en feed-forward laag. Bij decoders is daarnaast ook een derde self-attention laag aanwezig, waar de representaties van een encoder kunnen worden ingebracht. Wanneer decoders en encoders samenwerken in een Transformer model is er sprake van een encoder-decoder Transformer, waarvan de architectuur te vinden is in figuur 2.2.

Attention mechanisme

Het self-attention mechanisme, dat werd geïntroduceerd in de paper “Attention is all you need” van Vaswani e.a. (2017), zorgt ervoor dat eerdere delen van een input kunnen beïnvloeden hoe de rest van de input wordt geïnterpreteerd.

**Figuur (2.3)**

Visualisatie Attentie-mechanisme

Bijvoorbeeld in de zin 'Wij wandelen op het pad' en 'De pad zit in het meer' heeft het woord 'pad' telkens een verschillende betekenis, ofwel een smalle weg of wel een klein dier. 'Pad' is namelijk een homoniem, een woord met dezelfde uitspraak en spelling maar met een verschillende betekenis. Voor mensen is het eenvoudig om te bepalen welke van de twee bedoeld wordt aan de hand van de context. Bijvoorbeeld het woord 'meer' duidt voor mensen aan dat het wellicht over het dier gaat dat met water geassocieerd wordt. Voor een computer is dit heel wat lastiger, omdat het niet direct kan zien welke woorden er belangrijk zijn om de juiste betekenis te bepalen.

Het Attention mechanisme zal de numerieke representatie van het woord aanpassen door middel van de woorden rondom. Als men een embedding ziet als een vierkant met in de ene hoek de smalle weg en in de tegenoverstaande hoek het dier als numerieke representaties, start het woord 'pad' ergens middenin. Wanneer het woord 'wandelen' wordt teruggevonden in de zin zal dit de vector van 'pad' meer naar de hoek van 'wandelpad' 'trekken'.

Dit onderzoek was zo impactvol, dat de meeste AI modellen vandaag Transformers gebruiken, waaronder ook LLM's zoals ChatGPT.

2.5. Large Language Models

Large Language Models of LLM's zijn enorme taalmodellen die complexe taken kunnen uitvoeren zoals het genereren van poëzie of fungeren als een chatbot. Ze worden getraind op enorme hoeveelheden data, in het geval van GPT-3 op grote delen van het Internet. (OpenAI, 2022) Hieruit wordt een basismodel verkregen. Deze stap heet Generative Pre-training.

Daarna wordt het model verder gefinetuned met de hulp van menselijke AI trainers. Zij zorgen ervoor dat het model zich meer zal gedragen als een werkende chatbot.

De LLM werd immers getraind op internet-data die niet altijd het soort antwoorden bevat die men wilt verkrijgen van een chatbot die gebaseerd is op vraag en antwoord. In deze stap genaamd Supervised Finetuning spelen mensen de kant van zowel de vraagsteller als het model. Op deze manier leert de AI hoe het best antwoordt op input. Er wordt er ook voor gezorgd dat prompts die criminele of slechte bedoelingen hebben worden geweigerd.

Na deze stappen kan het model gepubliceerd worden. Het is dan aan de gebruiker om er de beste performantie uit te halen. Dit kan op twee manieren: fine-tuning en prompt engineering.

Fine-Tuning

Één van de manieren om een LLM beter te doen presteren voor een specifieke applicatie is fine tuning. Bij fine tuning wordt het eerder getrainde LLM nogmaals getraind maar met een veel kleinere dataset die aansluit bij het domein waarbij het model zal worden ingezet. Deze nieuwe trainingsdata bevat vaak een reeks inputs samen met de output die verwacht wordt. Hierna zal een menselijke trainer nogmaals het model testen en bij de gemaakte fouten nogmaals een nieuwe dataset opstellen waarmee het model getraind wordt. Dit is een iteratief proces waardoor het model telkens beter wordt. Volgens OpenAI (2024) zijn dit de verschillende use-cases om finetuning te gebruiken:

- Het bepalen van de stijl, toon, format of andere kwalitatieve aspecten
- Verbetering van de betrouwbaarheid bij het produceren van een gewenste output
- Het corrigeren van fouten bij het volgen van complexe aanwijzingen
- Veel randgevallen op specifieke manieren afhandelen
- Het uitvoeren van een nieuwe vaardigheid of taak die moeilijk in een prompt te verwoorden is

Prompt Engineering

Prompt Engineering is een nieuw idee waarbij de input voor een LLM zo goed mogelijk geformuleerd wordt zodat het model precies weet wat er verwacht wordt en betere resultaten kan geven.

Er bestaan verschillende modellen op de markt, waaronder betalende modellen maar ook open source modellen.

2.5.1. Propriëtaire LLM's

Propriëtaire LLM's zijn LLM's die ontwikkeld en onderhouden worden door een bepaald bedrijf. Deze modellen zijn meestal enkel betalend te gebruiken.

GPT-3.5 (OpenAI)

GPT-3.5 is werd ontwikkeld door de non-profit organisatie OpenAI en is de opvolger van GPT en GPT-2. Het is gebaseerd op decoder-only transformer architectuur. Ondertussen bestaat er ook een vierde versie van deze LLM genaamd GPT-4 maar deze is enkel gelimiteerd en betalend beschikbaar voor een maandelijkse som van 20 dollar. De meest gebruikte versie van GPT-3.5 is GPT-3.5 Turbo. De chatbot ChatGPT is gebouwd op een gefinetuned versie van deze LLM. Omdat een LLM niet zomaar met mensen kan praten werd het getraind tot chatbot door op grote hoeveelheden voorbeelden van communicatie te trainen tot het model het wenselijke gedrag vertoont.

Hoewel OpenAI een non-profit organisatie is, is de API enkel betalend te gebruiken. Tijdens het schrijven van deze bachelorproef bedroegen deze prijzen 0.5 USD per miljoen tokens voor input en 1.5 USD per miljoen tokens voor output. GPT-4 Turbo presteert beter dan GPT 3.5 Turbo maar is ook duurder, met een kost van 10 USD per miljoen tokens input en 30 USD per miljoen tokens voor output.

De webpagina over tokenisatie van OpenAI¹ definieert tokens als volgt: “Tokens zijn niet gelijk aan woorden, maar zijn veel voorkomende reeksen tekens die in een tekstset voorkomen. De modellen leren de statistische relaties tussen deze tokens begrijpen en blinken uit in het produceren van het volgende token in een reeks tokens”.

De huidige prijzen kunnen geraadpleegd worden op de website van OpenAI². Het gebruik van ChatGPT is gratis maar deze chatbot kan zelf niet in applicaties geïntegreerd worden. Dit kan enkel met de API. Wel kunnen er initiële tests mee worden uitgevoerd.

Gemini (Google)

Gemini is een multimodaal LLM ontwikkeld door Google. Voorheen stond dit model bekend als Bard. Multimodaal wil zeggen dan het met verschillende inputs en outputs kan werken, van audio, tekst en foto's, tot zelfs videobestanden. (Pichai & Hassabis, 2023).

Gemini heeft ook een API maar deze is nog niet beschikbaar in België. De enige manier om hiertoe toegang te krijgen is via een aparte service genaamd Vertex AI, wat het lastiger maakt om er mee te werken.

Llama 3 (Meta)

Llama 3 is de opvolger van Llama 2. Het model wordt ontwikkeld door Meta, het bedrijf achter onder andere *Facebook* en *Instagram*. Llama staat voor *Large Language Model Meta AI*.

Llama 3 werd aangekondigd op 18 april 2024, in de blogpost “Introducing Meta Llama 3: The most capable openly available LLM to date”³. De AI-assistent ‘Meta AI’

¹<https://platform.openai.com/tokenizer>

²<https://openai.com/api/pricing>

³<https://ai.meta.com/blog/meta-llama-3/>

maakt ook gebruik van het Llama 3 model, maar deze is nog niet beschikbaar in België.

2.5.2. Open Source LLM's

Open Source wil zeggen dat de broncode van het stuk software voor iedereen gratis toegankelijk moet zijn, en vrij mag aangepast en verspreid worden. Deze modellen zijn op zich gratis om te gebruiken, maar kunnen toch een kost met zich meedragen indien ze extern op een server *gehost* moeten worden via een cloud-service. Kleinere modellen kunnen ook lokaal op een eigen computer draaien, waardoor ze volledig gratis zijn in gebruik.

BERT

BERT is een techniek voor NLP pre-training ontwikkeld door Google in 2018. Het is een acroniem voor 'Bidirectional Encoder Representations from Transformers'. Het verbetert op vorige modellen omdat BERT de allereerste bidirectionele en *unsupervised* taalrepresentatie is. (Devlin & Chang, 2018)

BERT is een voorbeeld van een encoder-only transformer.

Bij de pre-training van GPT-modellen wordt enkel van links naar rechts tokens voorspeld. BERT gebruikt een nieuwe techniek genaamd "masked language model" of MLM. Hierbij worden willekeurig enkele tokens van de input gemaskeerd, en is het doel om het originele vocabularium-id van het gemaskeerde woord te voorspellen, gebaseerd enkel op de context van het woord. Hierdoor kan zowel de linkse als de rechtse context van het gemaskeerde woord samengenomen worden. (Devlin e.a., 2019)

Daarnaast is het ook volledig *open-source*, waardoor onderzoekers zelf hun eigen versies van BERT kunnen ontwikkelen. Zo is er bijvoorbeeld een eentalig BERT model dat gespecialiseerd is in de Nederlandse taal, genaamd BERTje. (de Vries e.a., 2019)

2.5.3. Vergelijking kwaliteit, snelheid en prijs

Het bedrijf Artificial Analysis voert onafhankelijk *benchmarks* uit voor LLM's, en publiceert de resultaten op hun website⁴. Dit onderzoek is dus een goede basis om de verschillende modellen op de markt te vergelijken. Het toont grafieken die de prijs, snelheid en kwaliteit van de LLM's vergelijken. Uit deze data werd besloten dat GPT-3.5 Turbo het meest geschikt lijkt voor dit onderzoek, vanwege de goede prijs-kwaliteit ratio en hoe toegankelijk het is.

⁴<https://artificialanalysis.ai/>

3

Methodologie

3.1. Verzameling van de testdata

Een van de bevindingen van het voorgaande onderzoek was dat er meer data nodig was om een juiste analyse uit te kunnen voeren. Van co-promotor J. Campens was toestemming verkregen om de data die bij een experiment binnen de opleiding verpleegkunde werd verzameld ook te gebruiken bij het schrijven van deze bachelorproef.

3.1.1. Opnames

Deze data werd als volgt verzameld.

Tijdens een middagpauze op bepaalde dagen werden een aantal vrijwillige studenten meegenomen naar het zorglabo om daar deel te nemen aan het experiment. Eens binnen, kreeg de student een microfoon opgespeld waarmee alle conversaties worden opgenomen.

Eerst werd het doel van het experiment kort uitgelegd. Daarna kreeg de student de opdracht om zichzelf kort voor te stellen. Het doel hiervan was om een basis te leggen van de normale spraak van de deelnemende student. Daar kan dan later de andere data mee vergeleken worden.

De volgende stap was om de student een jongere patiënt te laten helpen om zich klaar te maken om naar een revalidatiecentrum te gaan. Alle gesprekken werden opgenomen met de eerder opgespelde microfoon. Na de jongere patiënt te hebben geholpen werd de opdracht herhaald maar deze keer met een 80-jarige patiënt. Ook hierbij werd alles opgenomen.

Bij beide patiënten moest de student helpen om hun jas en schoenen aan te doen, en hen begeleiden naar de deur, ofwel in een rolstoel ofwel met behulp van een rolator. Na beide patiënten te helpen eindigde het experiment en werd de opname stopgezet.

3.1.2. Verwerking

Het verzamelen van deze data vond plaats op 4 dagen, met 4 deelnemers op dag 1, 7 deelnemers op dag 2, 6 deelnemers op dag 3 en 8 deelnemers op dag 4. In totaal zijn dit 25 deelnemers met elk een mp4 bestand van ongeveer 6 minuten lang. De studenten verpleegkunde hebben in een excel-bestand een overzicht gemaakt van alle kenmerken per opname:

- Aantal tussenwerpsels (voilà, hè)
- Aantal verkleinwoorden
- Aantal herhalingen
- Aantal collectieve voornaamwoorden
- Aantal troetelnamen

Daarnaast geeft het document ook niet-tekstuele kenmerken aan zoals loudness, pitch en stiltepercentage. Deze cijfers verkregen de studenten via de applicatie die vorige jaren voor dit onderzoek werd ontwikkeld.

3.1.3. Transcripts

Tijdens het schrijven van deze bachelorproef werkte medestudent B. De Schryver parallel aan het verbeteren van het speech-to-text algoritme. Zij heeft alle mp4-bestanden handmatig getranscribeerd om uiteindelijk de output van het speech-to-text algoritme te kunnen vergelijken met de werkelijke inhoud van het video-fragment. Deze transcripts werden ook gebruikt in dit onderzoek om een basis te vormen voor tekstuele analyse.

3.2. Classifiers

Oorspronkelijk was het idee om een *classifier* te ontwerpen die vanuit de verschillende kenmerken van *elderspeak* kan bepalen of een transcriptie van een audiobestand *elderspeak* bevat of niet. Hierbij worden al de kenmerken of *features* in een tabel geplaatst waarna een AI wordt getraind om bepaalde kenmerken te gaan associëren met een klasse. In dit geval zijn dit de klassen 'elderspeak' en 'geen elderspeak'. Het model zou uiteindelijk een percentage kunnen geven op basis van de aanwezige kenmerken van elderspeak. Bijvoorbeeld zou een groot aantal herhalingen en verkleinwoorden in een hoger percentage resulteren.

Het probleem was echter dat om een AI model zoals hierboven beschreven wordt een enorme hoeveelheid data vergt om mee te trainen. Daarnaast is ook een grote hoeveelheid rekenkracht nodig tijdens het trainen van het model. Beide waren minder haalbaar voor deze bachelorproef, maar zijn zeker nog een mogelijkheid voor een vervolgonderzoek.

3.3. Large Language Models

Een mogelijk idee om dit probleem aan te pakken is om data te laten genereren door een LLM zoals GPT-3.5. Om de werking van GPT-3.5 te testen kan initieel gebruik gemaakt worden van de webapplicatie ChatGPT, omdat deze versie gratis is, en dus geschikt is om een idee te krijgen over hoe goed dit model werkt voor deze use-case. Volgens onderzoek is GPT ook geschikt om in te zetten voor verschillende NLP taken zoals *Sentiment Analysis*. Mogelijks kan het dus ook de kenmerken van Elderspeak detecteren in een tekstfragment. GPT-4 presteert beter dan het oudere model maar is ook duurder. Daarom wordt in dit onderzoek enkel GPT-3.5 getest. Indien de resultaten voldoen aan de verwachtingen kan de GPT-3.5 API geïmplementeerd worden in de huidige Elderspeak-applicatie.

3.4. Verbetering tekstanalyse

Indien GPT-3.5 er niet genoeg in slaagt om op een correcte manier verkleinwoorden, tussenwerpsels en collectieve voornaamwoorden te extraheren uit een tekstfragment wordt er naar een alternatieve manier gezocht om deze taak te volbrengen. Daarbij kunnen klassieke NLP technieken worden toegepast zoals *part-of-speech tagging*. De nieuwe methode wordt vergeleken met de versie die voorheen in de Elderspeak-applicatie was geïmplementeerd. De vergelijking gebeurt door middel van confusion-matrices die worden opgesteld met behulp van een *python-library*.

4

Resultaten

4.1. Resultaten GPT-3.5

Alle resultaten van het onderzoek naar GPT-3.5 zijn te vinden in de links in bijlage [B.2.](#)

4.1.1. Genereren van gesprekken

Na een aantal testen was duidelijk dat via de ChatGPT web-applicatie realistische teksten gegenereerd konden worden die plaatsvonden bijvoorbeeld in een zorg-situatie in een rusthuis. Wanneer in de prompt werd toegevoegd dat als output ook *Elderspeak* moet geïncorporeerd worden voegde het model kenmerken toe zoals verkleinwoorden en troetelnamen, zoals in de zin “Ziet eruit alsof u probeert uw schoentjes aan te doen, mevrouw de Vries. Heeft u hulp nodig daarbij?”.

Wanneer gevraagd werd om nog meer *Elderspeak* toe te voegen, antwoordde de AI met zinnen zoals “Ah, kijk eens aan, mevrouw de Vries, bent u daar bezig met uw schattige schoentjes? Zal ik een handje helpen, liefje?”.

Er kon ook de opdracht gegeven worden om een gesprek over een bepaalde situatie te genereren. De prompt “De patiënt moet geholpen worden om aangekleed te worden” leverde zinnen op zoals “Zo, mevrouw de Vries, laten we vandaag dat prachtige jurkje aantrekken. Zal ik u even een handje helpen, schatje”.

Dit alles zonder dat het model specifiek verteld werd wat de kenmerken van *Elderspeak* zijn. De gegenereerde resultaten maken duidelijk gebruik van verkleinwoorden, troetelnamen en neerbuigende taal, hoewel die niet specifiek in de prompts vermeld werden.

Hieruit kan geconcludeerd worden dat GPT-3.5 een goed idee heeft over wat *Elderspeak* inhoudt.

4.1.2. Detecteren van *Elderspeak* in zinnen

Natuurlijk was de vraag, als een LLM voorbeelden van *elderspeak* kan genereren, kan het dit dan ook niet herkennen? Na testen met voorbeeldzinnen bleek dat dit ook mogelijk was met de juiste prompt.

GPT-3.5 herkende het gebruik van collectieve voornaamwoorden in korte zinnen. Er leek al wat moeilijkheid te bestaan met het detecteren van verkleinwoorden. Zo werd het woord 'daarbij' onjuist als verkleinwoord aangeduid en werd bij 'stoofpotjes' aangegeven dat het geen verkleinwoord is. Er was een hallucinatie van het model te zien waarbij het 'stoofpotjes' zelf verkleint tot 'stoofpotjetjes', wat geen correct Nederlands is.

Daarnaast gaf GPT-3.5 zelf alternatieve zinnen aan die respectvoller overkomen. Het leek ook subtielere vormen van *Elderspeak* te herkennen, zoals zinnen die mogelijks veronderstellen dat de oudere persoon hulp nodig heeft of de autonomie verminderen.

Indien geen *Elderspeak* aanwezig is werd dit ook op een correcte manier aangegeven. Op de prompt "is deze zin *elderspeak*: Natuurlijk, neem uw tijd. Als u toch assistentie nodig heeft, laat het me dan gewoon weten." gaf het model het volgende antwoord: "Nee, de gegeven zin is niet *elderspeak*. De toon is respectvol en beleefd, en er is geen gebruik gemaakt van verkleinwoorden of overdreven informeel taalgebruik. De zin drukt begrip uit voor de behoefte aan tijd en biedt hulp aan zonder neerbuigend te zijn. Het is eerder een beleefde en ondersteunende manier van communiceren."

Wanneer aan het model een lijst van zinnen werd doorgegeven met de opdracht om er degene uit te halen die *Elderspeak* bevatten, leek dit ook goed te gaan. Het gaf de redenen aan waarom het bepaalde dat een zin *Elderspeak* zou kunnen zijn, waardoor een gebruiker zelf hierover kan reflecteren. Sommige zaken leken niet helemaal correct, zoals het aangeven van een gesprek over het weer als *Elderspeak*, of het gebruik van uitroeptekens als betuttelend.

Bij één van de testen werd handmatig aan een lijst van 20 zinnen bij 5 ervan het tussenwerpsel 'hè' achteraan toegevoegd. Dit werd herhaald maar ditmaal met het woord 'hè' bij andere zinnen dan voorheen.

In het eerste geval haalde GPT-3.5 er alle zinnen uit waaraan het tussenwerpsel werd toegevoegd. Het model gaf ook het volgende aan: "Het gebruik van 'he' aan het einde van zinnen is informeel en kan als betuttelend worden ervaren." Er werd in de prompt niet aangegeven dat de taak was om het woord 'hè' te vinden. Omdat er in ChatGPT gebruik wordt gemaakt van *transformers* kan het taken uitvoeren op basis van de context van het gesprek, in dit geval het detecteren van *Elderspeak*. Ook bij de tweede test werden alle zinnen met 'hè' eruit gehaald, alsook 2 andere zinnen die beschouwd worden als betuttelend.

Het model gaf zelf ook aan dat het afhangt van de context: "Over het algemeen is *elderspeak* subjectief en kan het afhangen van de context en de relatie tussen

de spreker en de luisteraar. Wat voor de een als ondersteunend wordt beschouwd, kan voor de ander als betuttelend worden ervaren.”

Dit alles zonder dat GPT-3.5 specifiek getraind is om *Elderspeak* te detecteren, en zonder *fine-tuning*.

Echter was het belangrijkste deel van dit onderzoek om de kenmerken van *elderspeak* te detecteren, en de bestaande detectie-algoritmes vanuit het onderzoek van Van Gussem te verbeteren, waarna een algemener, overkoepelend resultaat optioneel was.

4.1.3. Detecteren van verkleinwoorden

Eerst werd een lijst van honderd zelfstandige naamwoorden gegenereerd met behulp van GPT-3.5. Daarna werd een aantal van deze woorden in de lijst handmatig een verkleinwoord gevormd. Tot slot kreeg het model de opdracht om de verkleinwoorden uit de lijst te extraheren.

In de eerste test stonden er 11 verkleinwoorden in de lijst. Het resultaat dat werd teruggegeven is een lijst van 19 woorden in totaal, waarvan er acht geen verkleinwoorden waren. Het model haalde 10 van de 11 verkleinwoorden er correct uit. Het woord dat niet correct werd teruggegeven is ‘Puzzelstukje’. Daarnaast werd het woord ‘zonsondergangetje’ tweemaal herhaald, terwijl het maar eenmaal in de lijst stond. Deze test werd nogmaals herhaald maar met een verschillende woordenlijst, en ditmaal achttien verkleinwoorden in de lijst. Veertien van de verkleinwoorden werden correct aangeduid bij deze test. De woorden die niet aanwezig waren in het resultaat, zijn ‘puzzelstukje’, ‘kasteeltje’, ‘olifantje’ en ‘vosje’. ‘Microfoon’ was ook terug te vinden in het antwoord, wat geen verkleinwoord is.

4.1.4. Detecteren van woordsoorten in transcripties

De resultaten van deze test zijn beschikbaar via de github-link die te vinden is in bijlage B.2.

Bij de eerste testen werd gevraagd om verkleinwoorden te halen uit één enkele zin, die komt uit de transcripties van het experiment beschreven in hoofdstuk 3.

GPT-3.5 leek hierbij regelmatig zelf verkleinwoorden te vormen die niet in de oorspronkelijke prompt staan.

De prompt “geef enkel de verkleinwoorden in de volgende zin: Zit die direct goed? Fantastisch, oké. Moet ik ik u schoenen, ah tis met een ritsje, ideaal.”

gaf het volgende resultaat: “zitje, schoentjes, ritsje”.

‘Zitje’ en ‘schoentjes’ stonden niet in de zin, wel ‘zit’ en ‘schoenen’.

Bij latere testen werden meerdere woordsoorten tegelijk opgevraagd, zoals verkleinwoorden, troetelnamen, collectieve voornaamwoorden en tussenwerpsels.

Aan de prompt kan ook het formaat van de gewenste output worden toegevoegd, in dit geval *json*. Er werd ook geëxperimenteerd met het opvragen van de complexiteit van de tekst, of het gehalte aan *Elderspeak*, maar hier leek weinig consistentie

in te zitten. Exact dezelfde tekst kreeg over verschillende testen telkens een andere waarde. Hier werd dus niet verder mee gewerkt in latere testen.

Hieronder een voorbeeld van hoe de resultaten eruitzagen in het gevraagde *json*-formaat. Dit voorbeeld is het resultaat van test met nummer 89 op GitHub¹.

```
{
  "verkleinwoorden met aantal": {
    "plaatsnemen": 1,
    "handtas": 1,
    "jasje": 1,
    "lab": 1,
    "schouderhoogte": 1,
    "broekzak": 1
  },
  "tussenwerpsels met aantal": {
    "All right": 2,
    "Yes": 1,
    "Goe": 2,
    "Oké": 13,
    "Ja": 20,
    "Voilà": 3,
    "Moh": 1,
    "Ah": 1
  },
  "troetelnamen met aantal": {},
  "collectieve voornaamwoorden met aantal": {
    "we": 1,
    "ons": 0
  }
}
```

Over de hele lijn had het model het lastig met verkleinwoorden. Het woord ‘schoenen’ kwam regelmatig voor als resultaat hoewel dit geen verkleinwoord is, ofwel werd er ‘schoentjes’ van gemaakt hoewel dit niet in deze vorm in de tekst staat.

De oorspronkelijke bedoeling voor deze test was om voor dit model tot een geschikte prompt te komen, en hiermee numerieke resultaten te vergelijken aan de hand van *confusion-matrices* en procentuele accuraatheid. Omdat er duidelijke fouten zichtbaar waren bij eender welke prompt, werd besloten dat deze oplossing nooit beter zou presteren dan wat reeds geïmplementeerd was in de bestaande *Elderspeak*-applicatie. Daarom werd geopteerd voor een ander soort AI-model om tot een beter presterend resultaat te komen, en hiervan wel de accuraatheid numeriek te analyseren.

4.2. Part of Speech Tagging

Om het detecteren van de verschillende woordsoorten te verbeteren wordt er getracht een *Part of Speech (POS) tagger* te gebruiken. Dit is een vorm van *Natural*

¹https://github.com/IneVdB/ElderspeakTests/blob/main/GPT_results/gpt_89_2.png

Language Processing waarbij voor elk woord in een zin het Part of Speech wordt toegekend.

Een *open source library* die hiervoor kan gebruikt worden is spaCy. Deze library beschikt over een vooraf getraind statistisch model waarmee het teksten kan analyseren. De pipelines voor de Nederlandse taal hebben een ingebouwde POS-tagger die verkleinwoorden, collectieve voornaamwoorden en tussenwerpsels kan aanduiden op basis van statistische waarschijnlijkheid en de context van de woorden.

De verkleinwoorden krijgen een tag met 'dim' erin wat staat voor 'diminutief' ofwel verkleinwoord. Door enkel de woorden te extraheren die volgens de spaCy-pipeline verkleinwoorden zijn kunnen alle verkleinwoorden uit een tekst worden gehaald. Hetzelfde geldt voor tussenwerpsels en collectieve voornaamwoorden.

4.2.1. Onderzochte woordsoorten

Hierbij een korte definitie van alle onderzochte woordsoorten:

Verkleinwoorden

Een verkleinwoord is over het algemeen een zelfstandig naamwoord met een bepaald achtervoegsel waardoor het veelal als vertederend of gezellig wordt gezien, maar kan het ook een neerbuigende of zelfs spottende connotatie geven. Mogelijke achtervoegsels zijn '-je', '-tje', '-kje', '-pje' en '-etje' (Cohen, 1984). In het Vlaams worden verkleinwoorden ook gevormd met het achtervoegsel '-ke' zoals in 'bolleke'. De spaCy-tagger geeft deze woorden een tag die eindigt op 'dim' wat staat voor 'diminutief', oftewel verkleinwoord. Een voorbeeld is de tag 'N|soort|mv|dim'.

Tussenwerpsels

Volgens van der Sijs (2001) zijn tussenwerpsels woorden die buiten de grammaticale structuur van de zin vallen. Daarnaast hebben ze geen syntactische verbindingsmogelijkheden, in plaats daarvan vormen tussenwerpsels op zichzelf een uitroepende zin.

In spaCy krijgen deze woorden de tag 'TSW' toegekend.

Collectieve voornaamwoorden

Collectieve voornaamwoorden zijn onder andere persoonlijke voornaamwoorden in de eerste persoon meervoud. Hieronder vallen de woorden 'we', 'wij' en 'ons'. Daarnaast is het woord 'onze', een niet zelfstandige bezittelijk voornaamwoord van de eerste persoon, ook een collectief voornaamwoord.

Door spaCy wordt aan deze woorden een tag toegekend die eindigt op '1|mv'.

Dit slaat op eerste persoon meervoud. Een voorbeeld is 'VNW|pr|pron|obl|vol|1|mv'.

Alle verschillende tags die spaCy kan toekennen zijn terug te vinden op de site van

de Nederlandstalige modellen².

4.2.2. Resultaat en vergelijking

Om de resultaten van de oude oplossing en de nieuwe implementatie met spaCy te kunnen vergelijken is manueel de aanduiding van deze kenmerken gebeurd in een apart bestand. Daarmee werd dan de data die de twee algoritmes als output geven vergeleken, en werd een score van accuraatheid verkregen als percentage.

Testmethode

In onderzoek naar classifieer-algoritmes gebruikt men hier doorgaans confusion-matrices voor. Dit is een tabelvorm die weergeeft hoeveel van de kenmerken juist werden aangeduid. Daarbij zijn 4 verschillende types terug te vinden:

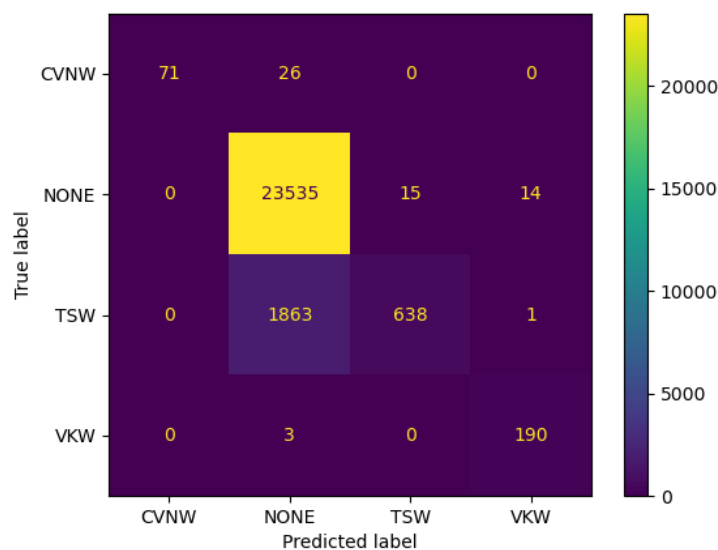
- 'False Postitives' of vals positieven: het algoritme geeft aan dat er een kenmerk aanwezig is maar dit is niet zo.
- 'True Positives' of juist positieven: het algoritme geeft correct aan dat het kenmerk aanwezig is.
- 'False Negatives' of vals negatieven: het algoritme geeft aan dat het kenmerk niet aanwezig is maar dat is het in werkelijkheid wel.
- 'True Negatives' of juist negatieven: het algoritme duidt aan dat het kenmerk niet aanwezig is en dat is het in werkelijkheid ook niet.

Voor elk woord in de gegeven transcriptie geeft het programma een label aan. De mogelijke labels zijn:

- VKW: Het woord is een verkleinwoord.
- CVNW: Het woord is een collectief voornaamwoord
- TSW: Het woord is een tussenwerpsel
- NONE: Het woord past in geen van de bovenstaande categorieën.

In het onderzoek van De Gussem (2022) werd reeds de accuraatheid getest van het algoritme dat hij schreef voor zijn bachelorproef. Echter gebeurde dit enkel op audio-fragmenten waardoor de kwaliteit van de text-to-speech implementatie ook meespeelde. Daarnaast werd er enkel getest of er al dan niet verkleinwoorden, tussenwerpsels of collectieve voornaamwoorden voorkomen in een mp3-bestand, en niet of deze correct werden aangeduid. Om dit op een betere manier te kunnen vergelijken met hetgeen wat voor deze bachelorproef tot stand is gebracht, moesten ook nieuwe testen voor het oude programma worden geschreven.

²<https://spacy.io/models/nl>

**Figuur (4.1)**

Accuraatheid oud programma

In de versie van De Gussem werd een andere, zelfgeschreven *tokenizer* gebruikt, wat ervoor zorgde dat het aantal tokens niet overeenkwam met het aantal tokens in de output van spaCy. Om een confusion-matrix te kunnen opstellen, wordt elk woord 1 op 1 vergeleken. Daarom moest dit wel overeenkomen.

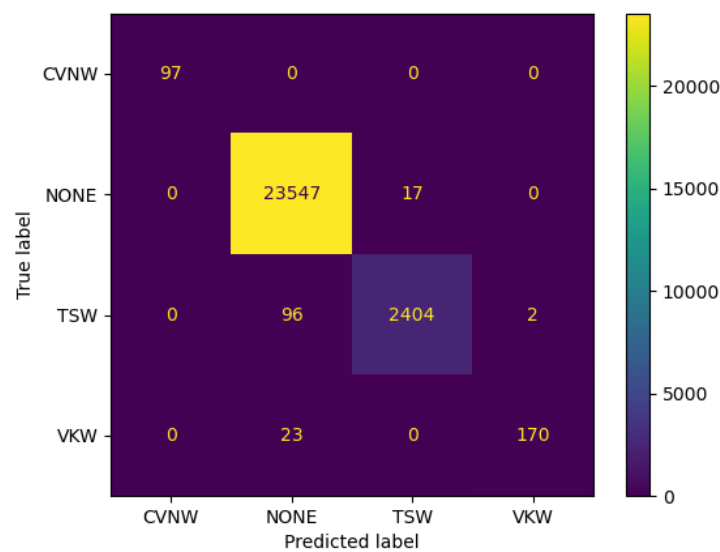
Om deze test te kunnen uitvoeren maakte het oude programma om deze reden ook gebruik van het tokenisatie-algoritme dat gebundeld zit in de spaCy-library. Alle broncode en resultaten van deze tests kunnen teruggevonden worden op GitHub³.

Er werd voor dit onderzoek gekozen voor het *large* model 'nl_core_news_lg' omdat het accurater is dan 'nl_core_news_sm' (*small*) of 'nl_core_news_md' (*medium*). Dit wel ten koste van de snelheid, maar accuraatheid is voor deze toepassing belangrijker.

Accuraatheid oud programma

De confusion-matrix voor het oude programma is te vinden in figuur 4.1. De accuraatheid die berekend werd bedraagt 92,71 procent. In de tabel is duidelijk dat het oude programma er minder goed in slaagde om alle tussenwerpsels correct aan te duiden. 1863 tussenwerpsels werden als 'NONE' gelabeld. Daarnaast duidde het ook 15 gewone woorden aan als tussenwerpsels, en waren 26 collectieve voornaamwoorden niet aangeduid. 14 gewone woorden werden foutief als verkleinwoord gelabeld. Dit gaat bijvoorbeeld over woorden zoals de naam 'Ilke' en 'moeje' die wel de juiste vorm hebben, maar geen echte verkleinwoorden zijn.

³<https://github.com/IneVdB/ElderspeakTests>

**Figuur (4.2)**

Accuraatheid nieuw programma

Accuraatheid nieuw programma

De confusion-matrix voor het nieuwe, verbeterde programma dat gebruik maakt van *POS-tagging* is te vinden in figuur 4.2. De berekende accuraatheid bedraagt 99,48 procent. Dit is dus duidelijk een verbetering ten opzichte van het vorige resultaat. Vanuit de *confusion-matrix* kan besloten worden dat alle collectieve voor-naamwoorden correct werden aangeduid. Daarnaast werden slechts 17 'gewone' woorden door het programma foutief aangeduid als een tussenwerpsel. Het grootste punt van verbetering is te vinden bij tussenwerpsels die niet werden aangeduid, en dus de tag 'NONE' kregen. Dit waren er in totaal 96. Er waren ook 2 tussenwerpsels die volgens het programma verkleinwoorden waren. 170 verkleinwoorden in alle transcripties waren correct aangeduid, maar er waren er ook 23 die niet werden herkend. Na analyse bleek dit vooral te gaan over dialectvormen van verkleinwoorden, d.w.z. woorden zoals 'efkes' en 'taske'. Om deze reden presteerde het oude algoritme beter op vlak van verkleinwoorden, het duidde 190 verkleinwoorden correct aan.

In bijlage B.3 staan alle resultaten van alle transcripties voor zowel het oude als het verbeterde programma.

4.3. Verbetering Lay-out

4.3.1. Beschrijving functionaliteit

De versie van de *Elderspeak*-detectie applicatie die het resultaat was van het onderzoek van De Gussem (2022) en Daems (2023) heeft de volgende functionaliteit (zie ook figuur 4.3):

Gewone spraak uploaden of opnemen



(HOGENT, 2022)

Spreek in hoe je tegen deze vrienden zou praten:

Choose File 1-03 (stude...011414.mp3)

standaard audio opnemen

Elder spraak uploaden of opnemen



(HOGENT, 2022)

Spreek in hoe je tegen deze oude vrouw zou praten:

Choose File 1-03 (stude...011418.mp3)

elder audio opnemen

▶ 0:00 / 1:40 🔊 ⋮

☒ Extract text

Verwerk audio

Pitch: 264.65
Loudness: -16.38
Pitch Filtered: 257.18
Loudness Filtered: -17.59
Stilte t.o.v totaal: 2.2 / 172.7
Stilte percentage: 1.27

Pitch: 300.74
Loudness: -25.01
Pitch Filtered: 317.08
Loudness Filtered: -27.41
Stilte t.o.v totaal: 4.3 / 100.5
Stilte percentage: 4.25
Lager of niet significant hoger
Stiller of niet significant luider

Wat heb je gezegd?

Hallo Google ik ben klaar aandoen want het is een beetje is koud ja we gaan eventjes misschien pakken makkelijkste zijn ja Moet ik helpen of kan je het zelf eens vriendelijk dat je eigenlijk dat gevraagd ja dat is het hey Blijkbaar als je ouder wordt moet er wat je nog kunt zeker doen hey tuurlijk niet meer hebben met dat centrum nee ik heb niks nodig Oké kijk gaan wij met de rollator of gaan we met de rolstoel van vandaag Ik ga hier al die voetjes naar binnen klappen Jazeker

Verkleinwoorden:

hallo google ik ben klaar aandoen want het is een **beetje** is koud ja we gaan **eventjes** misschien pakken makkelijkste zijn ja moet ik helpen of kan je het zelf eens vriendelijk dat je eigenlijk dat gevraagd ja dat is het hey blijkbaar als je ouder wordt moet er wat je nog kunt zeker doen hey tuurlijk niet meer hebben met dat centrum nee ik heb niks nodig oke kijk gaan wij met de rollator of gaan we met de rolstoel van vandaag ik ga hier al die **voetjes** naar binnen klappen jazeker

Figuur (4.3)Oude *Elderspeak*-detectie applicatie

Er zijn twee kolommen aanwezig, één voor het gesprek met een jongere persoon en één bedoeld voor het gesprek met een oudere persoon. De bedoeling is bij elke kolom een geluids- of videobestand op te laden. Daarnaast bestaat ook de mogelijkheid om in de applicatie zelf audio op te nemen. De twee bestanden worden geanalyseerd op basis van audiokenmerken zoals toonhoogte en stilte-percentages. Er wordt daarbij een vergelijking gemaakt tussen de twee bestanden. Indien de optie “extract text” is aangevinkt, maakt de applicatie ook een automatische transcriptie op waarin per kenmerk de woorden in rode tekst zijn aangeduid.

4.3.2. Aanpassingen

Er werd door co-promotor J. Campens aangegeven dat de automatische vergelijking geen belangrijke rol speelt bij het gebruik van de applicatie. Om de lay-out van de applicatie beter te laten aansluiten bij waarvoor het wordt ingezet, werd geopteerd om één enkele kolom te behouden in plaats van twee. Indien een gebruiker twee bestanden met elkaar wil vergelijken, kunnen ze op deze manier apart worden opgeladen.

De optie om spraak op te nemen binnen de applicatie werd ook verwijderd, omdat de analyse meestal gebeurt op reeds bestaande bestanden.

Om de tekstanalyse beter te visualiseren, wordt slechts eenmaal de transcriptie getoond, met daarin alle mogelijke *Elderspeak*-kenmerken gemarkeerd in kleur. Bovenaan de transcriptie staat een opsomming van het aantal voorkomens per kenmerk en in welke kleur ze zijn aangeduid in de tekst.

Tot slot werd een tekstveld toegevoegd waarin een manueel een transcriptie kan worden ingevoerd waar analyse op kan worden toegepast, zodat de applicatie nog steeds kan gebruikt worden ook indien de automatische transcriptie geen bruikbare resultaten oplevert.

Deze verbeterde lay-out is toegevoegd in figuur 4.4.

De broncode van de applicatie is ter beschikking gesteld op GitHub⁴, zodat er in een mogelijk vervolgonderzoek op verder gebouwd kan worden.

⁴<https://github.com/IneVdB/ElderspeakDetector>

Sprak uploaden

Choose File

1-03 (studen...011556.mp4)

☒ Extract text

Transcriptie invoeren (optioneel)

0:00 / 2:21

Verwerk audio

Pitch: 353.24

Loudness: -19.68

Pitch Filtered: 319.08

Loudness Filtered: -21.89

Stille t.o.v totaal: 5.1 / 141.3

Stille percentage: 3.63

Transcriptie:

Speech-To-Text Transcriptie

Aantal Tussenwerpsels gebruikt: 46

Aantal verkleinwoorden gebruikt: 1

Aantal Collectieve Voornaamwoorden gebruikt: 5

Tussenwerpsels worden in groen aangeduid, Verkleinwoorden worden in geel aangeduid, Collectieve voornaamwoorden worden in roze aangeduid

Ja, ik ben Greta. Ja Hallo Hallo Ik ben Emma. Ik kom nu begeleiden. Ja Jij bent Greta. Ik ben Greta. Hallo Ik ben Greta. Hallo Dat vind ik. Dat jij welkom delt. Ja Ja Goed. Moet ik jou helpen met je schoenen aan te doen? Of lukt dat allemaal aan te doen? Ja Mijn schoenen, dat zou wel helpen. Ja En mijn vrouw, ik kan je helpen met je schoenen. Ja Ja Ja En ik ben een muntje beperkt met een arm. Ah, oe. Ja Ik heb een zee in mijn steking met een arm. Oef Dus mijn hand is gevoelig. En daarmee is het er. Ah dat is niet het meest plezierige. Ah jammer. Dat is het eigenlijk. Wat is er? Hé Janne. Laten we ons kunnen laten verwinnen. Ah ja Natuurlijk. Als je je zei, dan kan je het gewoon niet. Als je er mogelijk in je zei, dan gaan we dat doen. Zo, dank je. Dank je. Goed. En het is ook van de eerste keren dat je naar daar gaat. Ja ja Ik heb hem nog nooit. Ik heb hem altijd dezelfde kunnen bedruipen, maar nu. Ah ja De middelen die er zijn, moeten me gebruiken. Natuurlijk. Ja Ik vind dat niet zo ambitant dat je soms zo nu wat meer help nodig hebt. Nee Ik vind dat als er de Europese moeten z'n varen. Ja En als je het niet nodig hebt, moeten we er ook je gebruik van maken. Zelfs gebeuren en doen, want dan kan je het zeker opleven. Ja Want dat groe kunt, zeker oplevertijdelijk wij. Ik weet dat je in leeftijd komt, dat er wat beperkingen komen. Op allerlei vlak. Ja ja Dat is eigenlijk aan je om er zoveel mogelijk nog uit te halen. Wat er uit te halen is. Natuurlijk. Ja ook. Dat zou ik ook wel doen. Ja En als ik hem er ga, dan ga ik hem dan. Ik ga het niet meer maken, maar ik wil wel voorstellen dat het soms wel makkelijk is. Als je zo moet kunnen laten helpen en zo. Het moeilijkste is aan vaten. Ja Dat het zelf niet met z'n groe kunt. Dat is het moeilijkste. Oké, ik had dat niet in de leven. Altijd alles goed gekund. Je had alles gedaan gelijk getwijd. Ja En nu moet het klaten. Rebeuren, rond u. Oké, ja. Dat is goed. Dat moet dan varen. Ja ja En dan gaat dat willen. Ja, waarschijnlijk. Ja, ja. Goed. Ik weet niet hoe ze dan nu gaan doen. Ik heb het idee dat ik ga naar de andere kant. Dat was misschien wel. Ja Ja dank je wel. Dag

Figuur (4.4)

Nieuwe Elderspeak-detectie applicatie

5

Vervolgonderzoek

Omdat de goede werking van het automatisch transcriberen van audio-bestanden cruciaal is om tekstanalyse uit te kunnen voeren, wordt in dit hoofdstuk kort een mogelijke oplossing beschreven.

Het speech-to-text algoritme Whisper bleek niet helemaal geschikt volgens het parallelle onderzoek van De Schryver, dus werd na afloop van het hoofdonderzoek van deze scriptie nog kort bestudeerd hoe dit verbeterd zou kunnen worden. Om tot een volledig resultaat te komen zal er wellicht nog een vervolgonderzoek over moeten gebeuren. Dit hoofdstuk dient als aanzet hiervoor.

5.1. Speaker Diarization

Bij *Speaker Diarization* wordt automatisch bepaald welke spreker op welk moment aan het woord is in een audiofragment. Dit is nuttig voor de *Elderspeak*-applicatie omdat het in de praktijk steeds gaat over situaties waar meerdere personen aan het woord zijn, waarvan minstens een patiënt en een zorgverlener. Het is niet van belang om te onderzoeken of de patiënt zelf *Elderspeak* gebruikt. Met *Speaker Diarization* kan dus uit de audio enkel de stukken gehaald worden waar de zorgverlener aan het woord is. Daarna kan enkel op de spraak van de zorgverlener analyse worden uitgevoerd.

5.2. Alternatief model

Er werd kort naar alternatieve speech-to-text modellen gezocht, waarbij features, prijs en accuraatheid de belangrijkste factoren zijn. Artificial Analysis¹ heeft ook een pagina die speech-to-text modellen vergelijkt. Dit gaat enkel over transcriberen in het Engels maar geeft toch een goed beeld van de andere factoren. Op de web-

¹<https://artificialanalysis.ai/speech-to-text>

site van EdenAI² kunnen tevens tegelijk verschillende modellen vergeleken worden door ze eenzelfde bestand te laten transcriberen. EdenAI geeft dan een overzicht van alle transcripties.

5.2.1. Assembly AI

AssemblyAI heeft volgens Artificial Analysis een lage word error rate (7%), gemiddelde snelheid van 43 seconden audio-transcriptie per seconde en een kost van 6,2 USD per 1000 minuten aan audio. In werkelijkheid is het grotendeels gratis te gebruiken omdat elk nieuw account start met 100 uur aan gratis transcripties³.

Daarnaast gaf het visueel de beste resultaten in EdenAI. De transcriptie van AssemblyAI transcribeert de volledige tekst waar sommige andere modellen hele stukken overslaan, en het heeft zo goed als geen onbestaande woorden wat bij Whisper bijvoorbeeld wel het geval is.

AssemblyAI heeft ook ingebouwde *Speaker Diarization* zonder extra kost. Wanneer deze functie wordt ingeschakeld, kent het model aan elke zin toe welke spreker aan het woord was. Een andere mogelijkheid van dit model is om aan elk gesproken woord een tijdstempel toe te kennen. Op deze manier kan bepaald worden op welk moment in het audiofragment een zin start of eindigt. Dit kan nuttig zijn bij verdere analyse van de geluidskenmerken van *Nursery Tone*, omdat het verhogen van de toonhoogte op het einde van een zin hier een kenmerk van is.

5.3. Resultaten

Tot slot werd kort onderzocht wat de accuraatheid is van AssemblyAI ten opzichte van de twee eerder onderzochte modellen, namelijk Google Speech Recognition API en Whisper. De broncode van deze test is beschikbaar op GitHub⁴. De videofragmenten waarop getest werd zijn niet publiek beschikbaar wegens privacyoverwegingen maar kunnen opgevraagd worden via e-mail.

De transcripties werden geanalyseerd via de volgende drie metrieken (Morris e.a., 2004) met behulp van de python-library *jiwer*:

- Word Error Rate (WER): Het aantal fouten in de transcriptie ten opzichte van het totaal aantal woorden.
- Match Error Rate (MER): Het percentage van woordovereenkomsten die fouten zijn.
- Word Information Lost (WIL): Een schatting van informatie die verloren ging in de transcriptie.

Bij elk van deze metrieken stelt een lagere waarde een betere accuraatheid voor.

²<https://www.edenai.co/>

³<https://www.assemblyai.com/pricing>

⁴<https://github.com/IneVdB/ElderspeakTests/tree/main/speech-to-text-tests>

De opnames van de eerste vier studenten die deelnamen aan het experiment van de opleiding geneeskunde werden door elk model getranscribeerd voor deze test, wat in totaal acht mp4-bestanden zijn. Met deze methode kan ook de rest van de bestanden gebruikt worden om een beter resultaat te verkrijgen indien een vervolgonderzoek uitgevoerd wordt, maar voor deze initiële test was een kleinere sample-size aangewezen wegens tijdslimitaties.

Tabel 5.1: Accuraatheid Speech-to-Text modellen

Model	WER	MER	WIL
AssemblyAI	0.524	0.513	0.682
Whisper Small	0.653	0.614	0.812
Google Speech Recognition	0.779	0.777	0.850

Uit deze resultaten kan geconcludeerd worden dat AssemblyAI beter presteert dan Whisper Small en Google Speech Recognition API voor deze use-case, en dit model dus kan ingezet worden bij een vervolgonderzoek dat gebruik kan maken van *Speaker Diarization* en tijdsstempels zoals beschreven. Mogelijk zijn er tegen dan vernieuwingen gebeurd dus het zal nog steeds nuttig zijn om de vernieuwingen op het gebied van speech-to-text op te volgen.

6

Conclusie

6.1. Bijdrage van dit onderzoek

Het gebruik van een POS-tagger om tekstuele kenmerken aan te duiden in een transcriptie vormt duidelijk een verbetering ten opzichte van wat er in voorgaande versies van de *Elderspeak*-applicatie gebruikt werd. Zo ging het percentage van correct aangeduide woorden van 92,71% naar 99,48%. Dit komt omdat er bij de POS tagger ook rekening wordt gehouden met de context van het woord en niet enkel of het woord al dan niet in een vooraf opgestelde lijst staat of een bepaald achtervoegsel heeft.

Daarom is de nieuwe versie zeker een verbetering, maar het is nog niet helemaal perfect. het spaCy model heeft het lastig met woorden dat het waarschijnlijk nog niet veel heeft gezien, zoals dialectvormen van verkleinwoorden. Dit kan mogelijk verholpen worden door het te combineren met een vorm van het oude algoritme dat specifiek op zoek gaat naar woorden op '-ke'.

De lay-out van de applicatie werd ook verbeterd om beter aan te sluiten bij het gebruiksdoel.

6.2. Gebruik van LLM's

OpenAI bracht op het einde van de bachelorproef-periode een nieuwe versie van hun GPT-model uit, wat er mogelijk wel in slaagt om correct de nodige tekstanalyse uit te voeren. Large Language Models zijn snel evoluerend en de mogelijkheden van de taken die ze kunnen volbrengen breiden steeds verder uit. Het kan dus nog nuttig zijn om in de toekomst een gelijkaardig onderzoek uit te voeren maar met een verbeterde versie van GPT of een ander LLM, en hier verder op in te gaan met meer geavanceerde technieken zoals fine-tuning.

Voor het onderzoek in deze scriptie werden geen betrouwbare resultaten verkregen met het GPT-3.5 model dat beschikbaar is via de webapplicatie van ChatGPT,

zelfs na verschillende *prompt-engineering* technieken toe te passen.

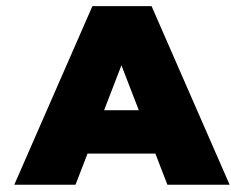
Het nieuwe GPT-4o model dat bekend werd gemaakt op 13 mei 2024 lijkt het na enkele korte tests beter te doen wat betreft het detecteren van verkleinwoorden, maar er was niet genoeg tijd over om hier verder op in te gaan.

De vraag is echter of het wel nodig is om een geavanceerd model te gebruiken voor een eerder triviale taak zoals verkleinwoorden aanduiden in een tekst. Volgens de resultaten van dit onderzoek kan een minder uitgebreid model dat zich enkel toespitst op tekstanalyse, zonder alle extra's, er ook in slagen om deze taak te volbrengen.

Daarom kan een LLM best ingezet worden voor een soort *sentiment analysis*, waarbij het model de gehele tekst analyseert en dan daaruit een conclusie trekt in hoeverre er *Elderspeak* aanwezig is, in plaats van het in te zetten enkel om de tekstuele kenmerken aan te duiden.

Ook kan het mogelijk herhalingen aanduiden, niet zozeer waar tweemaal exact hetzelfde gezegd wordt maar eerder twee verschillende uitspraken die dezelfde betekenis hebben. Het vaak herhalen van zaken, wat als betuttelend overkomt, is ook een kenmerk van *Elderspeak* waar in deze bachelorproef niet verder op werd ingegaan. Hierbij komt ambiguïteit aan te pas, alsook het belang van de context, wat LLM's geschikt maakt om deze taak uit te voeren, terwijl klassieke NLP-toepassingen minder over de juiste capaciteiten beschikken.

LLM's zijn een fantastische nieuwe technologie maar het is niet nodig om ze te gebruiken voor een taak waar een minder geavanceerd model even goed of zelfs beter presteert. Uit dit onderzoek kan dus geleerd worden dat een nieuwe, maar minder gekende of geteste technologie niet altijd de beste oplossing is om een probleem aan te pakken, wanneer de oude manier nog steeds werkt.



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.

Samenvatting

Dit onderzoek heeft als hoofddoel om het communiceren met ouderen in zorginstellingen te verbeteren aan de hand van technologie. Door betuttelend taalgebruik, zoals *Elderspeak*, automatisch met een computerprogramma te detecteren kunnen zorgverleners erop gewezen worden wat ze precies fout doen. Zo kunnen ze zelf aan de hand van deze informatie in de toekomst op een betere en respectvollere manier met patiënten omgaan. Omdat er steeds meer ouderen in rusthuizen terechtkomen kan een dergelijke oplossing een meerwaarde bieden aan onze maatschappij.

Het doel van dit eindwerk was om de bestaande *Elderspeak*-applicatie te verbeteren, waar reeds in vorige jaren binnen HOGENT onderzoek naar werd uitgevoerd. Deze applicatie zou idealiter worden ingezet binnen de opleiding verpleegkunde aan de HOGENT om studenten aan te leren wat *Elderspeak* is en hoe ze het gebruik ervan kunnen vermijden.

A.1. Introductie

Elderspeak is een actueel probleem dat zich voordoet bij zorgverlening aan ouderen. Ze worden door jonge mensen behandeld als kinderen wat hen minderwaardig kan laten voelen. (Louwet, 2017) Het doel van deze bachelorproef is om het voorgaande onderzoek verder uit te bouwen en het te verbeteren. Zo werd er nog niet genoeg data verzameld om tot een goed AI model te komen, en werd er enkel AI toegepast om de gesproken tekst om te zetten naar geschreven tekst.

De hoofdonderzoeksvraag is dus: Hoe kan AI worden ingezet om het model van de bachelorproef van De Gussem (2022) en Daems (2023) verder te verbeteren?

A.2. State-of-the-art

A.2.1. Natural Language Processing

Natural Language Processing (NLP) is een tak van het gebied van kunstmatige intelligentie en taalkunde, bedoeld om computers de uitspraken of woorden te laten begrijpen die in menselijke talen zijn geschreven. (Khurana e.a., 2022) De redenen om NLP in te zetten zijn als volgt (Gambäck, 2022):

- Computer Agents toestaan om met mensen te communiceren
- Computer Agents toelaten om informatie te verkrijgen uit geschreven tekst
- Mensen makkelijker laten communiceren met mensen

Het analyseren van tekst valt binnen het tweede punt, met name informatie verkrijgen.

Omdat taal kenmerken heeft die het analyseren moeilijk maken, is het steeds een uitdaging om taalmodellen volledig op punt te stellen. Natuurlijke taal is met name (Gambäck, 2022):

- Flexibel, constant evoluerend
- Onspecifiek en gebruikt in vele domeinen en toepassingen
- Redundant en verboos om te compenseren voor meerduidigheid

A.2.2. Classifiers

Classifiers zijn algoritmes die een gebruikersgedefinieerde klasse toekent aan een invoeritem. Daarom zijn ze ideaal om voor deze use case te gebruiken, de input is de gesproken tekst en het algoritme kent dan toe of het elderspeak is of niet. Het onderzoeken van de beste classifier voor deze use case vormt ook onderdeel van deze bachelorproef.

A.2.3. Voorgaand onderzoek

In het voorgaande onderzoek van De Gussem (2022) werd reeds een webapplicatie ontwikkeld die audiofragmenten met behulp van de Google Speech Recognition API gesproken tekst kan omzetten naar geschreven tekst. Op deze tekst wordt dan analyse uitgevoerd op basis van een aantal kenmerken van elderspeak, waaruit met enige nauwkeurigheid kan afgeleid worden of er al dan niet elderspeak aanwezig is in het fragment. Bij deze analyse wordt er nog geen gebruik gemaakt van AI, wat een verbeterpunt kan vormen.

A.3. Methodologie

Vooraleer er met het uitwerken van een PoC kan begonnen worden, zal er eerst meer data moeten verzameld worden. Er werden reeds 13 personen met 63 opnames in kaart gebracht, dit is echter niet genoeg om een AI classifier mee te trainen. Hiervoor zal contact opgenomen worden met een aantal zorginstellingen zodat er meer samples kunnen verkregen worden.

Het model zal gebruik maken van een aantal python-libraries die zijn ontwikkeld voor NLP en tekstanalyse. (Bengfort e.a., 2018) De belangrijkste hiervan is NLTK, Natural Language Toolkit.

Om het model te trainen zijn er features nodig, deze kunnen gemakkelijk afgeleid worden vanuit het voorgaande onderzoek, waarbij al verschillende methodes werden ontwikkeld om deze kenmerken te herkennen. De kenmerken die in beeld werden gebracht zijn:

- Verkleinwoorden
- Toonhoogte
- Stemvolume
- Herhalingen
- Collectieve Voornaamwoorden
- Tussenwerpsels

Deze data zal worden samengevoegd in een soort tabel, aan elk audiofragment worden de verschillende kenmerken toegekend als een abstracte score.

Op basis hiervan zal het model in staat kunnen zijn om te detecteren of er elderspeak aanwezig is, niet met een enkel kenmerk maar door ze allemaal te combineren. Dit zou een verbetering moeten zijn op het reeds bestaande model omdat het rekening houdt met het bredere beeld. Hierbij kan tevens ook een score worden toegekend naarmate het fragment meer of minder elderspeak bevat.

De verschillende classifiers kunnen vergeleken worden aan de hand van een percentage van juiste tegenover foute classificaties. De beste classifier zal degene zijn met het hoogste percentage juiste classificaties.

Zoals De Gussem (2022) vermeldde is het mogelijk dat de code om de aparte kenmerken te detecteren ook verbeterd of herschreven moet worden in een vervolgonderzoek. Het optimaliseren van de reeds bestaande functies zal dus ook behoren tot de scope van dit onderzoek. Zo kunnen er mogelijks NLP technieken worden ingezet om op een betere manier verkleinwoorden te detecteren. Eventueel kan hier ook een classifier worden ingezet die woorden beoordeelt op kenmerken van verkleinwoorden. Ook al is er veel meer NLP-onderzoek verricht over de Engelse taal dan in het Nederlands, werd er wellicht reeds een performante manier ontwikkeld om dit te realiseren.

A.4. Verwacht resultaat, conclusie

Ik verwacht dat door meer data te verzamelen en een beter model op te stellen met behulp van AI een betere accuraatheid kan bekomen worden om gesproken teksten te klassificeren als elderspeak.

B

Resultaten

B.1. Repositories

Onderzoek en testen

<https://github.com/IneVdB/ElderspeakTests>

Elderspeak-applicatie

<https://github.com/IneVdB/ElderspeakDetector>

B.2. Resultaten GPT-3.5

Test genereren van een gesprek in een rusthuis

<https://chat.openai.com/share/ad1b3f54-ba64-4b98-98f7-28d28a85f98e>

Test detectie van Elderspeak in zinnen

<https://chat.openai.com/share/e73ea226-904c-4420-aa61-f21305cd3726>

Test detectie van verkleinwoorden

<https://chat.openai.com/share/b6ad92e5-c687-43ff-aedc-f735c68dc159>

Test detectie tekstkenmerken Elderspeak in transcripties

https://github.com/IneVdB/ElderspeakTests/tree/main/GPT_results

B.3. Resultaten spaCy

BESTAND: transcripties\feb13_1.txt

----- NIEUW -----

```
{'vraagjes': 1, 'beetje': 8, 'voilà': 1, 'ritsje': 2, 'efkes': 1}
```

```
{'oké': 23, 'ja': 76, 'nee': 12, 'voilà': 7, 'ah': 9, 'hallo': 1,
  ↳ 'amai': 1, 'moh': 1, 'goh': 1, 'hé': 10, 'allez': 6}
{'we': 6}
```

```
----- OUD -----
```

```
{'vraagjes': 1, 'beetje': 8, 'ritsje': 2, 'efkes': 1}
{'ja': 75, 'voilà': 8, 'hallo': 1, 'hé': 9}
{'we': 6}
```

```
-----
BESTAND: transcripties\feb13_2.txt
```

```
----- NIEUW -----
```

```
{'vraagjes': 1, 'beetje': 4, 'weekje': 2, 'kwartiertje': 1,
  ↳ 'efkes': 4}
{'ja': 99, 'oké': 13, 'hallo': 2, 'nee': 14, 'allez': 3, 'voilà':
  ↳ 6, 'goh': 1, 'ah': 12, 'moh': 1, 'hé': 8, 'awel': 1, 'hup': 1}
{'we': 4}
```

```
----- OUD -----
```

```
{'vraagjes': 1, 'beetje': 4, 'weekje': 2, 'kwartiertje': 1,
  ↳ 'efkes': 5}
{'ja': 99, 'hallo': 2, 'eens': 1, 'voilà': 6, 'hé': 8}
{'we': 4}
```

```
-----
BESTAND: transcripties\feb13_3.txt
```

```
----- NIEUW -----
```

```
{'beetje': 5, 'efkes': 3, 'dekentje': 1}
{'ja': 70, 'ah': 8, 'oké': 13, 'hé': 9, 'hallo': 3, 'hoor': 2,
  ↳ 'voilà': 11, 'ziezo': 5, 'nee': 1}
{'we': 5}
```

```
----- OUD -----
```

```
{'efkes': 5, 'beetje': 5, 'dekentje': 1}
{'ja': 69, 'hé': 9, 'hallo': 3, 'zeg': 1, 'voilà': 11}
{'we': 5}
```

```
-----
BESTAND: transcripties\feb13_4.txt
```

```
----- NIEUW -----
```

```
{'efkes': 1, 'beetje': 4, 'vatjes': 1, 'eventjes': 1}
{'ja': 52, 'hé': 13, 'nee': 7, 'oké': 14, 'enfin': 1, 'ah': 1,
  ↳ 'hallo': 4, 'voilà': 3}
{'we': 8}
```

```
----- OUD -----
```

```
{'efkes': 2, 'beetje': 4, 'vatjes': 1, 'eventjes': 1, 'ietske': 1}
{'ja': 51, 'hé': 13, 'hallo': 4, 'voilà': 3}
```

```

{'we': 8}
-----
BESTAND: transcripties\feb20_1.txt
----- NIEUW -----
{'beetje': 7}
{'ja': 50, 'allez': 2, 'ah': 3, 'oké': 11, 'hallo': 4, 'hé': 8,
  ↳ 'nee': 3, 'goh': 1, 'voilà': 4}
{'we': 3, 'wij': 1, 'ons': 1}
----- OUD -----
{'efkes': 2, 'anke': 5, 'beetje': 7}
{'ja': 50, 'eens': 3, 'hallo': 4, 'hé': 8, 'zeg': 1, 'voilà': 4}
{'we': 3}
-----
BESTAND: transcripties\feb20_2.txt
----- NIEUW -----
{'dak': 1}
{'ja': 86, 'goh': 1, 'ah': 9, 'oké': 6, 'nee': 7, 'hallo': 4,
  ↳ 'voilà': 6, 'hé': 7, 'ziezo': 1}
{'we': 3}
----- OUD -----
{'klemmeke': 2}
{'ja': 80, 'hallo': 4, 'voilà': 6, 'hé': 7}
{'we': 3}
-----
BESTAND: transcripties\feb20_3.txt
----- NIEUW -----
{'beetje': 4, 'lokaaltje': 2}
{'ja': 60, 'oké': 11, 'ah': 4, 'awel': 2, 'voilà': 4, 'hé': 3,
  ↳ 'hallo': 4, 'nee': 2, 'goh': 1, 'ziezo': 1}
{'wij': 1}
----- OUD -----
{'beetje': 4, 'lokaaltje': 2, 'vrije': 1, 'ietske': 1}
{'ja': 60, 'voilà': 3, 'eens': 3, 'hé': 3, 'hallo': 4}
{}
-----
BESTAND: transcripties\feb20_4.txt
----- NIEUW -----
{'microotje': 1, 'klemmetje': 1, 'standaardvraagjes': 1, 'beetje':
  ↳ 6, 'schilderijtjes': 1}
{'voilà': 5, 'ja': 50, 'allez': 3, 'oké': 3, 'nee': 4, 'hé': 10,
  ↳ 'hallo': 3, 'moh': 1, 'ah': 1, 'mwa': 1}

```

```

{}
----- OUD -----
{'microotje': 1, 'klemmetje': 1, 'standaardvraagjes': 1, 'beetje':
  ↳ 6, 'schilderijtjes': 1, 'efkes': 1}
{'voilà': 5, 'ja': 50, 'eens': 2, 'hé': 10, 'hallo': 3}
{}
-----
BESTAND: transcripties\feb20_5.txt
----- NIEUW -----
{'klemmetje': 1, 'beetje': 9, 'efjes': 1, 'efkes': 1}
{'oké': 10, 'ja': 47, 'hé': 10, 'voilà': 15, 'ah': 2, 'goh': 1,
  ↳ 'hallo': 5, 'hoor': 1, 'bah': 1, 'hupsakee': 4, 'amai': 1,
  ↳ 'nee': 4}
{'we': 5}
----- OUD -----
{'klemmetje': 1, 'beetje': 9, 'efjes': 1, 'efkes': 1}
{'ja': 47, 'hé': 9, 'voilà': 15, 'hallo': 5}
{'we': 5}
-----
BESTAND: transcripties\feb20_6.txt
----- NIEUW -----
{'beetje': 3}
{'ja': 32, 'nee': 3, 'oké': 6, 'goh': 1, 'ah': 5, 'hallo': 4, 'hé':
  ↳ 4, 'voilà': 7}
{'we': 1, 'onze': 1}
----- OUD -----
{'klemmeke': 1, 'beetje': 3, 'rekske': 1}
{'ja': 32, 'hallo': 4, 'hé': 4, 'voilà': 6}
{'we': 1}
-----
BESTAND: transcripties\feb20_7.txt
----- NIEUW -----
{'efkes': 1, 'beetje': 4}
{'ja': 70, 'oké': 18, 'oh': 1, 'ah': 2, 'allez': 1, 'hallo': 3,
  ↳ 'hé': 8, 'voilà': 8, 'ziezo': 1, 'nee': 7, 'awel': 1}
{'we': 9, 'wij': 1, 'onze': 1}
----- OUD -----
{'efkes': 1, 'beetje': 4}
{'ja': 67, 'hallo': 3, 'hé': 7, 'voilà': 8, 'jaja': 1}
{'we': 9}
-----

```


BESTAND: transcripts\feb21_1.txt

----- NIEUW -----

```
{'teentje': 1, 'uh': 1, 'beetje': 2}
{'ja': 52, 'hé': 6, 'oké': 11, 'allez': 1, 'ah': 9, 'hallo': 4,
  ↳ 'oei': 2, 'oh': 1, 'amai': 1, 'voilà': 10, 'nee': 1}
{'we': 2, 'wij': 1}
```

----- OUD -----

```
{'efkes': 1, 'neelke': 1, 'teentje': 1, 'oranje': 1, 'beetje': 2}
{'ja': 51, 'hé': 6, 'hallo': 3, 'voilà': 9}
{'we': 2}
```

BESTAND: transcripts\feb21_2.txt

----- NIEUW -----

```
{'beetje': 7, 'deeltje': 1}
{'oké': 4, 'ja': 31, 'allez': 1, 'hé': 7, 'voilà': 18, 'nee': 5,
  ↳ 'oei': 2, 'ah': 2, 'hallo': 1, 'mwa': 1, 'oeie': 1, 'amai': 1}
{'we': 4}
```

----- OUD -----

```
{'beetje': 7, 'deeltje': 1, 'taske': 1}
{'ja': 30, 'hé': 7, 'voilà': 17, 'hallo': 1}
{'we': 4}
```

BESTAND: transcripts\feb21_3.txt

----- NIEUW -----

```
{'beetje': 6}
{'voilà': 8, 'oei': 1, 'ja': 41, 'oké': 5, 'ah': 4, 'awel': 1,
  ↳ 'hé': 6, 'nee': 7, 'allez': 2, 'och': 1, 'oh': 4, 'hallo': 2,
  ↳ 'ai': 1, 'wel': 1, 'sorry': 1, 'goh': 1}
{}
```

----- OUD -----

```
{'beetje': 6}
{'voilà': 8, 'ja': 38, 'hé': 6, 'jaja': 1, 'hallo': 2}
{}
```

BESTAND: transcripts\feb21_4.txt

----- NIEUW -----

```
{'beetje': 9, 'machientje': 1}
{'ja': 31, 'nee': 5, 'oké': 6, 'voilà': 6, 'ah': 3, 'allez': 1,
  ↳ 'hé': 5, 'moh': 1}
{'we': 1}
```

----- OUD -----

```
{'kijkenje': 1, 'efkes': 1, 'beetje': 9, 'machientje': 1}
{'ja': 32, 'eens': 1, 'voilà': 6, 'hé': 5, 'zeg': 1}
{'we': 1}
```

```
-----
BESTAND: transcripties\feb21_5.txt
```

```
----- NIEUW -----
{'beetje': 6}
{'ja': 28, 'allez': 1, 'oké': 11, 'hallo': 3, 'hé': 5, 'voilà': 9,
  ↳ 'oei': 3, 'ah': 4, 'ziezo': 2, 'hoor': 1}
{'we': 8}
```

```
----- OUD -----
{'karreke': 1, 'beetje': 6}
{'ja': 27, 'hallo': 3, 'hé': 5, 'voilà': 9}
{'we': 8}
```

```
-----
BESTAND: transcripties\feb21_6.txt
```

```
----- NIEUW -----
{'beetje': 3, 'efkes': 1}
{'ja': 33, 'allez': 1, 'ah': 1, 'oké': 4, 'hallo': 3, 'hé': 6,
  ↳ 'nee': 1, 'voilà': 1, 'goh': 1, 'awel': 1}
{'wij': 1, 'we': 1}
```

```
----- OUD -----
{'beetje': 3, 'efkes': 1}
{'ja': 30, 'hé': 6, 'voilà': 1, 'hallo': 2}
{'we': 1}
```

```
-----
BESTAND: transcripties\mar1_1.txt
```

```
----- NIEUW -----
{'beetje': 5, 'voetjes': 1}
{'hé': 7, 'ja': 40, 'ah': 1, 'oké': 4, 'hallo': 4, 'voilà': 10,
  ↳ 'nee': 3, 'ach': 1, 'awel': 3, 'oh': 1, 'moh': 1}
{'we': 4}
```

```
----- OUD -----
{'beetje': 5, 'voetjes': 1}
{'hé': 7, 'ja': 36, 'zeg': 1, 'hallo': 4, 'voilà': 10}
{'we': 4}
```

```
-----
BESTAND: transcripties\mar1_2.txt
```

```
----- NIEUW -----
{'beetje': 1}
```

```
{'ja': 81, 'voilà': 3, 'ah': 4, 'amai': 3, 'oké': 3, 'hallo': 5,
  ↳ 'hé': 19, 'nee': 10, 'hupsakee': 1, 'oh': 2, 'ma': 1, 'allez':
  ↳ 1, 'awel': 1, 'bah': 1}
{'we': 7, 'ons': 1}
```

----- OUD -----

```
{'ilke': 3, 'beetje': 1}
{'ja': 80, 'voilà': 3, 'hallo': 5, 'hé': 19}
{'we': 7}
```

BESTAND: transcripts\mar1_3.txt

----- NIEUW -----

```
{'beetje': 6}
{'oké': 11, 'ach': 1, 'ja': 50, 'ah': 6, 'goh': 2, 'hallo': 4,
  ↳ 'nee': 6, 'oei': 1, 'voilà': 3, 'hoor': 1, 'hé': 8, 'allez': 1}
{'wij': 1}
```

----- OUD -----

```
{'efkes': 1, 'beetje': 6}
{'ja': 49, 'hallo': 4, 'eens': 1, 'voilà': 3, 'hé': 8}
{}
```

BESTAND: transcripts\mar1_4.txt

----- NIEUW -----

```
{'bakje': 1, 'beetje': 6}
{'ja': 33, 'awel': 1, 'oké': 15, 'hallo': 4, 'nee': 1, 'bwa': 1,
  ↳ 'ah': 3, 'hé': 3, 'voilà': 7, 'oh': 2, 'och': 1}
{'we': 1}
```

----- OUD -----

```
{'bakje': 1, 'beetje': 6, 'efkes': 1}
{'ja': 33, 'hallo': 4, 'hé': 3, 'voilà': 7}
{'we': 1}
```

BESTAND: transcripts\mar1_5.txt

----- NIEUW -----

```
{'beetje': 6, 'klachtekes': 1}
{'ja': 59, 'oké': 10, 'da': 1, 'hoor': 1, 'awel': 3, 'hé': 11,
  ↳ 'ah': 10, 'nee': 12, 'hallo': 1, 'voilà': 7}
{'we': 5}
```

----- OUD -----

```
{'efkes': 1, 'beetje': 6, 'beeke': 2, 'klachtekes': 1}
{'ja': 57, 'hé': 11, 'hallo': 1, 'jaja': 1, 'voilà': 7}
{'we': 5}
```

```

-----
BESTAND: transcripts\mar1_6.txt
----- NIEUW -----
{'beetje': 9}
{'oké': 8, 'ja': 49, 'hoor': 1, 'allez': 3, 'hallo': 5, 'ah': 8,
  ↪ 'hé': 7, 'awel': 1}
{'ons': 1}
----- OUD -----
{'beetje': 9}
{'ja': 48, 'hallo': 5, 'hé': 7}
{}
-----
BESTAND: transcripts\mar1_7.txt
----- NIEUW -----
{'beetje': 6}
{'ja': 42, 'oké': 10, 'hé': 6, 'allez': 5, 'hallo': 5, 'hey': 1,
  ↪ 'ah': 10, 'voilà': 3, 'ai': 1, 'oei': 1, 'awel': 2}
{'we': 3}
----- OUD -----
{'beetje': 6}
{'ja': 42, 'hé': 5, 'hallo': 5, 'hey': 1, 'voilà': 3}
{'we': 3}
-----
BESTAND: transcripts\mar1_8.txt
----- NIEUW -----
{'beetje': 7}
{'ah': 13, 'ja': 78, 'nee': 5, 'oké': 7, 'hallo': 5, 'voilà': 4,
  ↪ 'alsjeblieft': 2, 'hé': 12, 'allez': 1, 'awel': 1, 'ma': 1,
  ↪ 'bah': 1, 'oef': 1, 'oh': 2, 'moete': 1}
{'we': 4, 'ons': 1, 'wij': 1}
----- OUD -----
{'moeje': 1, 'beetje': 7, 'oeije': 1}
{'ja': 78, 'eens': 1, 'hallo': 5, 'voilà': 4, 'hé': 12}
{'we': 4}
-----

```

Bibliografie

- Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer Normalization. <https://doi.org/10.48550/arXiv.1607.06450>
- Bengfort, B., Bilbro, R., & Ojeda, T. (2018, juni). *Applied Text Analysis with Python: Enabling Language Aware Data Products*. O'Reilly Media, Inc.
- Cohen, H. (1984). Huisje, boompje, beestje. *Onze Taal*. https://onzetaal.nl/uploads/editor/8409_Verkleinwoorden.pdf
- Daems, B. (2023). *Nursery Tone Monitor: Verbetering van Elderspeech-detectie door middel van ruisonderdrukking en stilteherkenning*. <https://catalogus.hogent.be/catalog/hog01:000752907>
- De Gussem, S. (2022). *Nursery Tone Monitor: detecteren van elderspeak via AI*. <https://catalogus.hogent.be/catalog/hog01:000747773>
- Devlin, J., & Chang, M.-W. (2018). *Open Sourcing BERT: State-of-the-Art Pre-training for Natural Language Processing*. Verkregen maart 19, 2024, van <https://research.google/blog/open-sourcing-bert-state-of-the-art-pre-training-for-natural-language-processing/>
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *North American Chapter of the Association for Computational Linguistics*.
- de Vries, W., van Cranenburgh, A., Bisazza, A., Caselli, T., Noord, G. v., & Nissim, M. (2019). BERTje: A Dutch BERT Model. <http://arxiv.org/abs/1912.09582>
- Duyck, J., Nevejan, H., Paul, J.-M., & Vandresse, M. (2022). 1,3 miljoen inwoners meer tegen 2070, vergeleken met 1,5 miljoen in de afgelopen 30 jaar. De coronacrisis heeft geen invloed op deze groei. *Flash*. https://www.plan.be/uploaded/documents/202202080742020.Persp_FLASH_DP21_12568_N.pdf
- Gambäck, B. (2022). *Cursus Intelligent Text Analytics and Language Understanding*.
- Khurana, D., Koli, A., Khatter, K., & Singh, S. (2022). Natural Language Processing: State of The Art, Current Trends and Challenges. *Multimedia Tools and Applications*, 82. <https://doi.org/10.1007/s11042-022-13428-4>
- Lipizzi, C., Verma, D., Korfiatis, G., Borrelli, D., Capela, F., Clifford, M., Desai, P., Giffin, R., Hespelt, S., Hoffenson, S., Kanchi, S., Khan, M., McDermott, T., Pepe, K., Ramirez-Marquez, J., Sa, P., Saremi, R., See Tao, H. Y., Shankar, R., & Yu, Z. (2019). Meshing Capability and Threat-based Science and Technology Resource Allocation. https://www.researchgate.net/publication/343323519_Meshing_

Capability_and_Threat-based_Science_and_Technology_ST_Resource_Allocation

- Louwet, K. (2017). *Infantilisering en betutteling van ouderen*. Verkregen november 1, 2023, van https://www.upckuleuven.be/sites/default/files/calendar_item/s5_dr_stefaan_soenen_karen_louwet_tessa_vanspauwen.pdf
- Malingan, N. (2023, mei 4). Verkregen mei 23, 2024, van <https://www.scaler.com/topics/nlp/transformer-encoder-decoder/>
- Merton, R. K. (1967). *Social theory and social structure*. Free Press.
- Morris, A., Maier, V., & Green, P. (2004). From WER and RIL to MER and WIL: improved evaluation measures for connected speech recognition. <https://doi.org/10.21437/Interspeech.2004-668>
- OpenAI. (2022). *Introducing ChatGPT*. Verkregen maart 14, 2024, van <https://openai.com/blog/chatgpt>
- OpenAI. (2024). *When to use fine tuning*. Verkregen maart 12, 2024, van <https://platform.openai.com/docs/guides/fine-tuning>
- Pichai, S., & Hassabis, D. (2023). *Introducing Gemini: Our largest and most capable AI model*. Verkregen maart 21, 2024, van <https://blog.google/technology/ai/google-gemini-ai/>
- Robins, R. H., & Crystal, D. (2024). *language*. Verkregen mei 16, 2024, van <https://www.britannica.com/topic/language>
- Shaw, C., & Gordon, J. (2021). Understanding Elderspeak: An Evolutionary Concept Analysis. *Innovation in Aging*, 5. <https://doi.org/10.1093/geroni/igab023>
- Smit, A., & van Vuuren, T. (2023). *Leeftijd is een getal, geen beperking: Verslag van een onderzoek naar age-isme in de arbeidsdeskundige praktijk. In opdracht van het Arbeidsdeskundig Kennis Centrum*. Arbeids Deskundig Kenniscentrum. <https://www.arbeidsdeskundigen.nl/akc/kennis/akc-cahiers/document/5451>
- van der Sijs, N. (2001). *Chronologisch woordenboek: De ouderdom en herkomst van onze woorden en betekenissen*. Het Taalfonds.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is All You Need. <https://arxiv.org/pdf/1706.03762.pdf>