Samen aanjagen van vernieuwing

Automatische spraakherkenning

Hoe kun je het inzetten voor onderwijsmateriaal?

Auteur(s): Arjan van Hessen
Versie: 1.0
Datum: 1 augustus 2020

Inhoudsopgave

1 Inleiding

SURF werkt aan een zoekportaal waarin je als docent of student open leermaterialen kunt delen en zoeken. Op 1 plek vind je daar een groot en gevarieerd aanbod van digitale leermaterialen, zoals kennisclips, PowerPoint-presentaties, pdf's, online e-modules en oefentoetsen. Ideaal voor docenten die op zoek zijn naar inhoudelijke vernieuwing of nieuwe werkvormen om de kwaliteit van leermaterialen te verbeteren.

Om goed te kunnen zoeken in al die materialen, maken we gebruik van full text search. Tekstmateriaal is vrij gemakkelijk te doorzoeken, maar hoe maak je video- en audiomateriaal doorzoekbaar. Om uit die materialen de tekst de extraheren, is automatische spraakherkenning nodig.

In dit rapport laat spraakherkenningsdeskundige Arjen van Hessen zien in hoeverre het mogelijk is om automatische spraakherkenning (automatic speech recognition, ASR) in te zetten voor het transcriberen van video- en audiomateriaal. De belangrijkste vragen die we hem stelden, zijn:

- Hoe goed werken de huidige technieken?
- Wat moet er gedaan worden om de resultaten van automatische spraakherkenning te verbeteren?
- Hoeveel inspanning kost dat?

Ben je (net als wij) benieuwd naar wat er komt kijken bij spraakherkenning en hoe het precies werkt? In dit rapport vind je een aantal tips, maar ook technische achtergronden.

Hoe is het rapport opgebouwd?

- In hoofdstuk 2 kijken we naar het bronmateriaal: hoe zorg je ervoor dat het audio- en videomateriaal dat getranscribeerd moet worden, optimaal is voorbereid voor automatische spraakherkenning?
- In hoofdstuk 3 lees je meer over de concepten die komen kijken bij automatische spraakherkenning: voor ASR heb je een taalmodel en een akoestisch model nodig.
- Het resultaat van ASR, de automatisch gegenereerde transcriptie dus, is niet perfect. Hoe je dat kunt corrigeren om tot een goede tekst te komen, lees je in hoofdstuk 4.
- De resultaten moeten ook gepresenteerd kunnen worden. Dat kan op verschillende manieren (denk aan ondertiteling). Meer daarover lees je in hoofdstuk 5.
- Een volgende stap is automatisch vertalen van de resultaten. Dit wordt behandeld in hoofdstuk 6.
- Tenslotte bevat hoofdstuk 7 een verdieping op automatische spraakherkenning: hoe ga je precies aan de slag met een taalmodel en een akoestisch model?

Voor ons eigen project hebben we veel gehad aan de informatie die Arjen van Hessen in dit rapport bijeengebracht heeft. We hopen dat het ook waardevol is voor projecten die jij aan je instelling uitvoert.

2 Bronmateriaal optimaliseren

Er zijn simpel gezegd twee soorten audiovisuele (AV) documenten: die al gemaakt zijn en die nog gemaakt moeten worden. Aan de kwaliteit van reeds gemaakte documenten kan niet veel meer gedaan worden. Voor het door mensen beluisteren (en bekijken) van het AV-document kan het zin hebben om bijvoorbeeld ruis te onderdrukken of de amplitude van het signaal te verhogen.

Voor automatische spraakherkenning (ASR) maakt dit in de regel niet veel uit. Het verdubbelen van de amplitude is niet veel meer dan een simpele bit-shift van alle waarden van het signaal maar de verhouding tussen de opeenvolgende waarden blijft volkomen gelijk waardoor ook de berekende parameters gelijk zijn. Het verwijderen van ticks, bromtonen (bijvoorbeeld 50 Hz) en klokgetik kan soms helpen.

2.1 Opnamen

2.1.1 Apparatuur

Bij het maken van nieuw materiaal is het wel verstandig om een aantal zaken in de gaten te houden.

Het beste kan een (eenvoudige) wired/wireless ‘ijzerdraad’-microfoon gebruikt worden (zie afbeelding). Het voordeel van deze microfoons is dat ze nauwelijks hinder geven aan de spreker(s) en ook nauwelijks zichtbaar zijn op de video. Door de constructie blijft de afstand tussen microfoon en mond constant waardoor de amplitude niet veranderd wanneer de spreker zijn/haar hoofd beweegt.

Hetzelfde kan bereikt worden met een Bluetooth microfoon van bijvoorbeeld Plantronics. Voordeel is dat veel van dit soort microfoons actieve ruisonderdrukking hebben. Er zijn twee microfoons (een op de mond gericht en een naar de buitenwereld. De twee signalen worden dan van elkaar afgetrokken waardoor er een erg rustig en stil signaal overblijft.

Als dit soort microfoons niet gebruikt kunnen worden, is een lapel-lavalier microfoon (met draad) een goede oplossing. De microfoon wordt met een clip op de kraag/overhemd/trui geklikt waardoor de afstand tussen microfoon en mond redelijk constant blijft. Voor video-opnamen is dit uitstekend omdat je de microfoon niet of nauwelijks ziet.

Als dit ook niet kan (te begrotelijk, niet beschikbaar) dan kan ook gebruik gemaakt worden van de ‘oortjes’ die met iedere smartphone worden meegeleverd. De oortjes zelf hebben geen toegevoegde waarde maar zorgen er wel voor dat de microfoon (in de draad) op een redelijk constante afstand tot de mond blijft zitten. Nadeel is dat bij het bewegen, de microfoon soms tegen de kleding schuurt hetgeen duidelijk hoorbaar kan zijn.

Wat echter niet gedaan moet worden, is het opnemen van een spreker door een smartphone gewoon op tafel te leggen en de ingebouwde microfoon gebruiken. Doordat de smartphone op tafel ligt zullen alle contacten met de tafel worden opgenomen (schuiven van objecten op de tafel, neerleggen van pennen of kopjes etc.). Bovendien weerkaatst het geluid aan het tafeloppervlak en dat is hoorbaar op de opnamen.

Als er al met alleen met een telefoon wordt opgenomen (zonder externe microfoons), dan moet de telefoon op een stapel zachte stoffen worden gelegd (muts, handdoek, trui) om de contactgeluiden zoveel mogelijk te vermijden.

Bovenstaande oplossingen gebruiken de computer of smartphone voor de geluidsopnamen. De eenvoudigste (maar dikwijls niet beste) oplossing is een standalone dictafoon (mono of stereo).

De meeste dictafoons hebben een of twee goede microfoon(s) waarmee in een rustige omgeving een gesprek goed kan worden opgenomen. Nadeel is echter dat de kanaalscheiding meestal niet goed genoeg is om de spraakherkenning op elk kanaal apart te doen (te veel overspraak).

2.1.2 Opnameparameters

Spraakherkenning wordt in de regel gedaan met een 16-16-1[1] geluidsfile in ongecomprimeerd wav-formaat. Voor een optimale weergave is het verstandig om in een hogere kwaliteit o te nemen, maar voor de herkenning maakt dit niets uit. Wat wel vermeden moet worden, is het opnemen in een sterk gecomprimeerd formaat. Voor het menselijk gehoor maakt compressie niet zoveel uit, maar voor de spraakherkenner wel.

De gemaakte audio-opnamen moeten getranscodeerd worden naar 16-16-1 voordat ze door de spraakherkenner gehaald kunnen worden.

2.2 Aantal sprekers

Het aantal sprekers dat moet worden opgenomen, bepaalt in zekere mate de opnameapparatuur. In het ideale geval wordt iedere spreker op een eigen kanaal opgenomen en is er weinig tot geen overspraak. Op die manier is het eenvoudig om te detecteren wie wanneer spreekt. En door ieder kanaal apart door de spraakherkenner te halen, is ook bekend wie wat wanneer zegt.

2.2.1 Eén spreker

Voor zover bekend is er bij de meeste kennisclips en opgenomen colleges, sprake van één spreker. In dat geval is het relatief eenvoudig: de spreker wordt zo goed mogelijk opgenomen en het achtergrondlawaai wordt zo veel mogelijk vermeden en/of eruitgefilterd.

Alle hierboven besproken opnameapparatuur is in principe geschikt.

[1] 16 kHz sampling frequency, 16-bit samples, 1 kanaal (mono).

2.2.2 Twee sprekers

Voor gesprekken met twee personen zijn er verschillende oplossingen. Bijna alle laptops en smartphones hebben de mogelijkheid om geluid op te nemen; maar meestal is dit slechts in mono en is er één input poort beschikbaar. Om toch met twee microfoons te kunnen opnemen kan een breakoutbox gebruikt worden. Daarin zitten twee input poorten die het signaal samenvoegen naar één kanaal en dat de computer/smartphone in sturen. Het opgenomen signaal komt dan van twee microfoons maar wordt samengevoegd en als een kanaal opgeslagen (mono). Ook hier moet dan later met de hand en/of middels software de sprekers gescheiden worden.

Om toch in stereo op te nemen (waarbij de ene spreker op het linker en de andere op het rechter kanaal worden opgenomen) moet er een extra apparaatje gebruikt worden. Voor de laptop zijn er verschillende oplossing (vanaf 40 euro). Voor de iPhone is er een breakoutbox die op de lightning poort wordt aangesloten.

2.2.3 Meerdere sprekers

Bij meerdere sprekers is dat anders. In het ideale geval wordt iedere spreker op een eigen kanaal opgenomen, is er weinig overspraak en wordt ieder kanaal apart door de herkenner gehaald. De herkenningsresultaten worden daarna weer bij elkaar gevoegd waarbij iedere spreker bijvoorbeeld een eigen kleur krijgt. In de werkelijkheid is dit lastig te realiseren bij meer dan 2 sprekers.

Fig. 1: Ideale opnamensituatie waarbij iedere spreker een eigen microfoon heeft en bekend is wie bij welke microfoon hoort.

Het gaat alleen in vergaderzalen die hiervoor speciaal zijn ingericht. Iedereen die iets wil zeggen drukt een knopje in waardoor de andere microfoons gemutet worden. Hierdoor is achteraf feilloos te bepalen wie wanneer sprak en is de opnamekwaliteit over het algemeen uitstekend.

Fig. 2: Meerkanaalsoplossing waarbij tot 8 verschillende microfoons gebruikt worden en elke microfoon een eigen kanaal heeft. Kan met bedrade en draadloze microfoons gebruikt worden.

Een andere oplossing is het gebruik van speciale (studio) hardware zoals de hier afgebeelde Tascam-recorder die 8 microfoons tegelijkertijd kan opnemen en het signaal via USB naar de computer stuurt. Maar het zijn prijzige apparaten, zeker omdat men ook speciale microfoons moet gebruiken.

Als er toch meerdere sprekers tegelijk moeten worden opgenomen en bovenstaande oplossing niet realiseerbaar is, dan kan gekozen worden voor een omnidirectionele 360-graden-tafelmicrofoon die midden tussen de sprekers gelegd wordt. De output is echter mono zodat later met de hand en/of middels software de verschillende sprekers uit elkaar gehaald moeten worden.

3 Automatische spraakherkenning

Spraakherkenning is de laatste jaren heel snel heel veel beter geworden. Er zijn meer data beschikbaar, de algoritmes zijn beter (deep neural networks) en de computers zijn sneller. Maar perfect is het niet en zal het waarschijnlijk ook nooit worden. Ook mensen zijn geen perfecte spraakherkenners, maar wij corrigeren onze imperfecte herkenning met kennis van de context. We weten waar het over gaat, en vullen de niet of slecht gehoorde informatie aan met wat we denken dat er gezegd werd.

Dankzij het gebruik van taalmodellen, doen spraakherkenners iets dergelijks, maar een taalmodel is een vooraf gebouwd statistisch model en dat is niet hetzelfde als kennis van de context: waar gaat de spraak over!

De ASR-fouten kunnen grofweg worden onderverdeeld in herkenningsfouten en out-of-vocabulary-fouten.

3.1 Herkenningsfouten

Herkenningsfouten worden onder andere veroorzaakt doordat de geluidsopname niet goed is, doordat er iemand anders doorheen spreekt, er achtergrondgeluid is, doordat de spreker ‘onduidelijk’ spreekt of wanneer de fonetische transcriptie van een woord verkeerd is. Bij een woord dat niet herkend wordt of dat als een ander woord herkend wordt, is er sprake van een herkenningsfout.

Out-of-vocabulary-fouten (OOV) zijn fouten die worden veroorzaakt doordat het te herkennen woord **niet** in het woordenboek staat en dus niet herkend kan worden. Moderne spraakherkenners kunnen veel woorden herkennen (de Nederlandse KALDI-herkenner kan 256.000 verschillende woorden herkennen), maar er zit dus een grens aan. Het huidige Nederlands kent ongeveer 1,3 miljoen woorden dus de meeste woorden kunnen niet herkend worden. Gelukkig komen die niet-te-herkennen woorden weinig voor en beschrijven de 256.000 woorden 98% van de Nederlandse spraak/teksten. Maar zeker bij ASR voor speciale contexten (bijvoorbeeld De medische context) kunnen OOV-fouten optreden.

Spraakherkenning rust op twee pijlers: een akoestisch model dat de fonemen (de betekenis onderscheidende klanken) van een taal bevat en een taalmodel dat de te herkennen woorden bevat en de statistische relatie tussen deze woorden.

3.2 Akoestisch model

De eerste stap bij het herkennen van spraak is het bepalen van de opeenvolgende betekenisvolle klanken in de spraak: de **fonemen**. Iedere taal heeft een min of meer vaste set betekenisvolle klanken waarmee de spraak wordt gevormd.

3.2.1 Berekenen fonemen

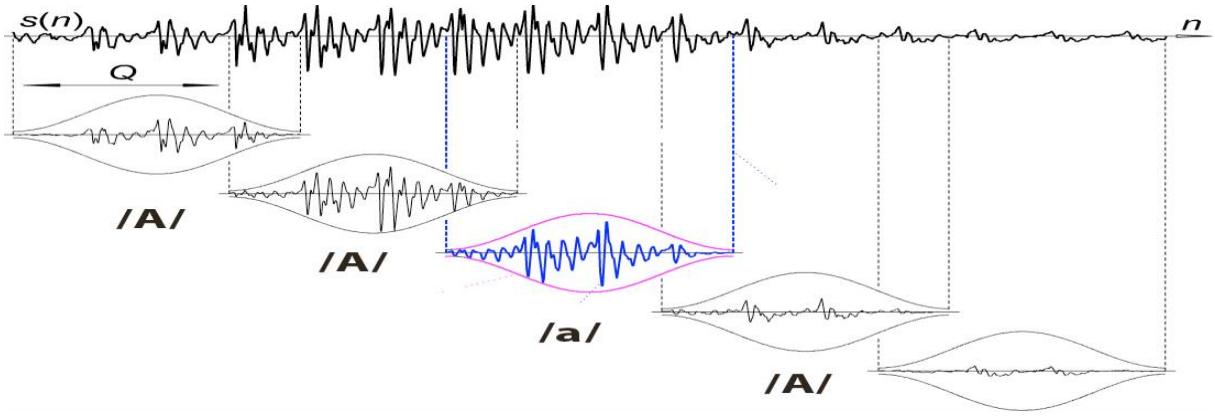


Fig. 3: Schematische weergave van het berekenen van de parameters van een audiosignaal waarbij het window steeds een stukje opschuift. 3 keer wordt het foneem /A/ (van man) herkend en 1 keer het foneem /a/ (van maan).

Van een signaal dat binnenkomt, worden de eerste 25 ms genomen en van die 25 ms audio worden een aantal parameters berekend. Dan schuift het signaal 10 ms op en worden er weer 25 ms signaal genomen waarvan opnieuw de parameters berekend worden. De parameters worden zo goed mogelijk vergeleken met parameters die van ieder foneem berekend zijn. Als de berekende parameters dan erg lijken op de parameters van foneem /A/ dan wordt dat stukje van 25 ms gezien als een /A/. Op die manier wordt van elk stukje van 25 ms het meest waarschijnlijke foneem berekend. Maar wanneer de uitgesproken klank bijvoorbeeld een Utrechtse /A/ is (met de /A/ van man) dan lijkt die /A/ erg op de /a/ (van maan). De rij fonemen kan dan een afwisseling zijn van opeenvolgende /A/'s en /a/'s. En als er net iets meer /a/'s voorkomen dan kan bijvoorbeeld maan herkend worden terwijl er man gezegd werd.

Ook kan het gebeuren dat een woord 'sloppy' wordt uitgesproken omdat de spreker weet dat er geen ander woord is waarmee het verward kan worden. Een aantal fonemen wordt dan niet uitgesproken maar de menselijke luisteraar 'weet' wat er bedoeld werd. Neem een zin als "ik werd gebeld door mijn verzekeringsmaatschappij". Verzekeringsmaatschappij kan heel erg sloppy worden uitgesproken, en toch zullen wij (Nederlandstalige) luisteraars verzekeringsmaatschappij 'horen'. Maar de computer ziet een reeks opeenvolgende fonemen die niet overeenkomen met de reeks die je verwacht bij de officiële uitspraak. Dan kan het snel misgaan en is er sprake van een herkenningsfout.

3.2.2 Categorische perceptie

De menselijke spraak bestaat uit oneindig veel verschillende klanken. Maar al die verschillende klanken worden per taal gegroepeerd in fonemen: betekenis**onderscheidende** klanken.
Een voorbeeld zijn de 3 Nederlandse fonemen /i/ (zoals in piet), /I/ (zoals in pit) en /E/ (zoals in pet). We kunnen een continuüm of reeks maken die van de /i/ naar de /I/ naar de /E/ gaat. Sprekers van het Nederlands zullen dan eerst verschillende /i/'s horen, dan verschillende /I/'s en tenslotte verschillende /E/'s.

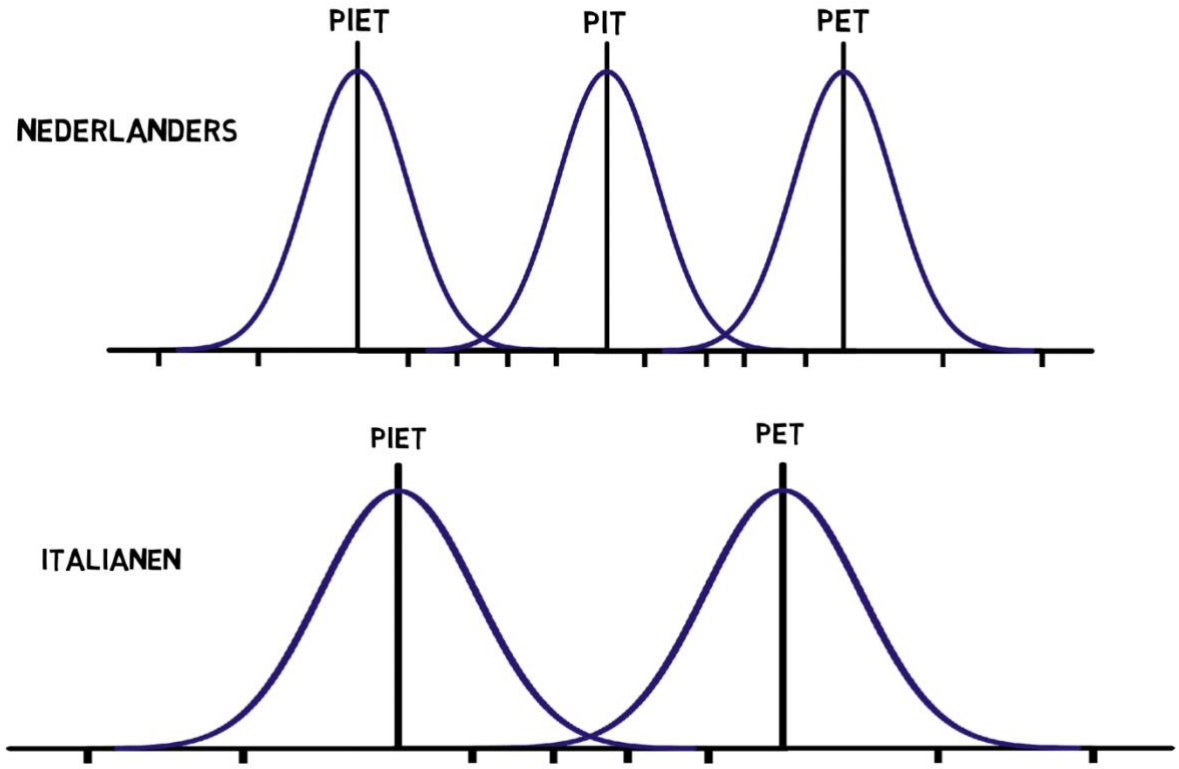


Fig. 4: Schematische weergave van categorische perceptie. Daar waar Nederlanders de /I/ van pit horen, horen Italianen soms een /i:/ (van piet), soms een /E/ (van pet).

Ergens halverwege de /i/ en de /I/ zal men in verwarring raken en soms een /i/ en soms een /I/ horen. Hetzelfde gebeurt halverwege de /I/ en de /E/. De verschillende realisaties van deze 3 klinkers zijn de **fonemen** ofwel door mensen gemaakte klanken. Ieder mens (in dit geval Nederlander) spreekt anders en zal dus een net iets andere uitspraak van de /i/ hebben. Maar ook dezelfde spreker zal steeds een /i/ net iets anders uitspreken. Maar zolang die uitgesproken /i/ redelijk in de buurt van de Nederlandse /i/ ligt, zal een Nederlander al die realisaties van de /i/ als een /i/ herkennen.

Voor een Italiaan ligt dat echter anders. Het Italiaans kent geen /I/ en in de hierboven beschreven reeks van /i/ -> /I/ -> /E/ zal een Italiaan slechts twee klanken horen: de /i/ (als in pizza) en de /E/ (als in del). Daar waar Nederlanders de /I/ horen, zal een Italiaan juist gaan twijfelen tussen de /i/ en de /E/.

Dezelfde reeks klanken worden door Nederlanders dus anders gegroepeerd dan door Italianen. Wij verdelen ze in drie categorieën en de Italianen in twee.

Iets soortgelijks zie je bij het Chinees waarin onze /r/ en de /l/ (dus twee fonemen) als een foneem worden gezien. Wij kennen echter één /E/ (als in bed) terwijl het Engels er twee kent: voor bed en bat.

3.2.3 Nederlandse fonemen

Met andere woorden: dezelfde klanken zullen door sprekers van verschillende talen, verschillend worden waargenomen. Nu zijn talen niet statisch en veranderen ze in de loop der tijd. Ze

veranderen doordat sprekers van de taal bestaande woorden anders gaan uitspreken en nieuwe woorden introduceren. Bovendien worden talen beïnvloed door vreemde talen: zeker als het contact tussen mensen die verschillende talen spreken intensiever wordt (handel, muziek, toerisme, internet). Zo gelden de klinker /9:/ (als in fr**eu**le) en de nasaal /a~/ (op het einde van restaur**ant**) als niet-Nederlands. Maar als ze vaak in Nederlandse spraak worden gebruikt, dan moeten ze deel uitmaken van de Nederlandse foneemset. Een goed overzicht van de fonemen van het Nederlands is te vinden op de SAMPA pagina van het UCL.

3.2.3.1 Medeklinkers

Plosieven: p b t d k (g):

Symbool	**Woord**	**Transcriptie**
p	pak	pAk
b	bak	bAk
t	tak	tAk
d	dak	dAk
k	kap	kAp
g	goal	go:l (alleen leen- of buitenlandse woorden)

Fricatieven: f v s z x (G) h:

Symbool	**Woord**	**Transcriptie**
f	fel	fEl
v	vel	vEl
s	sein	sEin
z	zijn	zEin
x	toch	tOx
G	goed	Gut (also xut)
h	hand	hAnt
Z	bagage	bAga:Z(@)
S	show	So:u

Sonoranten (nasalen, liquids en glides): m n N l r w j:

Symbool	Woord	Transcriptie
m	met	mEt
n	net	nEt
N	bang	bAN
l	land	lAnt
r	rand	rAnt
w	wit	wIt
j	ja	ja:

3.2.3.2 Klinkers

De Nederlandse klinkers vallen in twee klassen, 'gecontroleerd' (niet voorkomend in een gestresste lettergreep zonder een volgende medeklinker) en 'vrij'.

De gecontroleerde klinkers: I E A O Y @:

Symbool	Woord	Transcriptie
I	pit	pIt
E	pet	pEt
A	pat	pAt
O	pot	pOt
Y	put	pYt
@	geluk	G@-lYk

De 'vrije' klinkers bestaan uit:
4 monoftongen i y u a:
3 'mogelijke diftongen' e: 2: o:
3 'essentiële diftongen' Ei 9y Au

Symbool	Woord	Transcriptie
i	vier	vir
y	vuur	vyr
u	voer	vur

a:	naam	na:m

e:	veer	ve:r
2:	deur	d2:r
o:	voor	vo:r

Ei	fijn	fEin
9y	huis	h9ys
Au	goud	xAut

Er zijn bovendien 6 klinkercombinaties die soms ook als diftong worden gezien:

Symbool	**Woord**	**Transcriptie**
a:i	draai	dra:i
o:i	mooi	mo:i
ui	roeiboot	ruibo:t
iu	nieuw	niu
yu	duw	dyu
e:u	sneeuw	sne:u

Klinkers in leenwoorden

Symbool	**Woord**	**Transcriptie**
E:	crème	krE:m
9:	freule	fr9:l@
O:	roze	rO:z@

Dit resulteert in 47 Nederlandse fonemen.

3.2.4 Fonetische transcriptie

Om woorden te kunnen herkennen, moet de computer weten hoe ze (kunnen) worden uitgesproken. De meeste woorden hebben een (veel voorkomende) uitspraak maar een redelijk aantal woorden hebben er meer, zoals melk (melk of melluk). Ook wordt de uitspraak beïnvloed door het accent/dialect van de spreker. Zo wordt de finale schwa-n (als in lopen: /l o: p @ n/) in het westen zonder de finale /n/ uitgesproken (-> /l o: p @/) maar in het Noorden en Oosten juist zonder de finale schwa (-> /l o: p n/) die als het ware wordt ingeslikt.

Tenslotte hebben we een canonieke uitspraak (de officiële uitspraak volgens het Groene Boekje en/of Van Dale) en de realistische, dagelijkse uitspraak zoals de meeste Nederlanders het in gewone spraak uitspreken. Als voorbeeld kan gedacht worden aan de uitspraak van het woord verzekeringsmaatschappij. Officieel is dat /v E r - z 'e: - k @ - r "I N s - m a: t - s X A - p Ei/) maar zo spreekt (bijna) niemand dat uit. De /E/ wordt een schwa (/@/), de eerste /r/ wordt dikwijls niet uitgesproken, en de /z/ wordt dikwijls een /s/ (en de /v/ een /f/, zeker als de spreker uit Amsterdam komt).

Los van het effect van de herkomst van de spreker, is het bij een lang woord als verzekeringsmaatschappij ook niet zo belangrijk dat het woord 'correct' wordt uitgesproken: iedere Nederlander zal het woord uitgesproken als /f @ s e: k r I N s m a: t s X @ p Ei/ als verzekeringsmaatschappij herkennen. Er zijn immers geen akoestisch concurrerende woorden die bijna net zo klinken. De spreker kan het woord dus 'slordig' uitspreken zonder het risico te lopen dat een ander woord herkend wordt. Anders ligt het bij een woord als 'safe' (/s e: f/) en 'zeef' (/z e: f/) hoewel bij gelijke uitspraak de context wel duidelijk maakt welk van de twee bedoeld wordt.

3.2.5 G2P

Een spraakherkenner herkent in eerste instantie klanken en schat vervolgens welke woorden de waargenomen reeks klanken veroorzaakt zouden kunnen hebben (Bayesian Approach). En daarom moet de herkenner weten hoe de woorden uitgesproken kunnen worden. Een woord als **melk** kan worden uitgesproken als /m E l k/ of als /m E - l @ k/. En wanneer een van deze twee reeks klanken 'voorbij' komen, dan kan de herkenner besluiten dat de kans dat melk de waargenomen reeks klanken gegenereerd heeft, erg hoog is en dat er dus waarschijnlijk melk gezegd werd. Van alle 256.000 woorden die herkend moeten kunnen worden, moet dus bekend zijn hoe ze 'klinken'.

Om de juiste uitspraak van al deze woorden te krijgen, wordt gebruikgemaakt van een G2P (Grapheme-to-Phoneme) tool. Zo'n G2P-tool zet de woorden om in een reeks (taalafhankelijke) mogelijke fonemen.

De meeste G2P-engines combineren een woordenboek met een set regels. Een woordenboek heeft het voordeel dat de transcriptie door mensen gecontroleerd is en daarom meestal juist. De set regels zorgt ervoor dat nieuwe/onbekende woorden ook fonetisch getranscribeerd kunnen worden, maar de transcriptie kan fouten bevatten omdat het een 'vreemd' woord is. Voorbeelden van 'vreemde' woorden zijn bijvoorbeeld XS4ALL. Een 'normaal' Nederlands woord bevat geen cijfers en begint bovendien nooit met de combinatie van een X en een S en dus zal de G2P-engine hier in de spellingsmode gaan
(/i k s - E s - v i: r - a - E l - E l/).

3.2.5.1 Stress en syllabes

Officieel geeft de G2P-tool naast de fonemen ook de syllabegrenzen (lettergrepen) aan (-) en de primaire (‘) en secundaire (‘’) stress aan (de klemtonen) maar in de fonetische transcripties in dit document worden die meestal weggelaten. De meeste spraakherkenners (in tegenstelling tot TTS-systemen) gebruiken echter geen stress en syllabegrenzen en dus worden ze weggehaald.

3.2.5.2 (De-)compounding

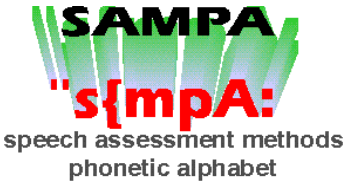
Omdat het Nederlands een ‘samenstellingstaal’ is, kunnen er on-the-fly nieuwe woorden gemaakt worden. Ze verscheen het woord ‘vuurwerkramp’ pas na de vuurwerkramp in Enschede (2001) in de Nederlandse kranten.
Meestal gaat de automatische transcriptie van nieuwe woorden wel goed omdat een goede G2P-engine eerst kijkt of een onbekend woord wellicht uit twee of meer bekende woorden bestaat. Als dat zo is, wordt de fonetische transcriptie van beide woorden genomen en worden die achter elkaar gezet. Het splitsen van samenstellingen kan soms lastig zijn, omdat de twee delen van de samenstelling met een tussen-s aan elkaar geplakt worden (beslissing + bevoegdheid => beslissing**s**bevoegdheid)

Abchazië
Ab hazen
Abc azen

Maar ook dit kan fout gaan. Zo werd in een eerdere versie van de Twente G2P-engine het woord ‘Abchazen’ (mensen uit Abchazië) gesplitst in ‘ABC’ en ‘hazen’ en ‘zeeroverschatten’ in ‘zeerovers’ en ‘chatten’.

3.2.6 SAMPA

De officiële fonetische transcripties worden gemaakt met zogeheten International Phonetic Alphabet (IPA). Omdat het lastig is om de transcripties in IPA weer te geven (je moet er het IPA-font voor downloaden en installeren) en je transcripties niet makkelijk in IPA kunt maken, is er een simpeler versie ontwikkeld: SAMPA. SAMPA (Speech Assessment Methods Phonetic Alphabet). De meeste spraakherkenners (waaronder de KALDI-herkenner) gebruiken de SAMPA-annotatie of een verrijkte versie hiervan.



3.2.6.1 Multiple transcripties

Om de mogelijkheid aan te geven dat er op een bepaalde plek in de fonetische transcriptie verschillende opties zijn, wordt het pipe-symbool (|) gebruikt.

Delft -> /`d E l (|@|Y) f t`/ leidt tot:

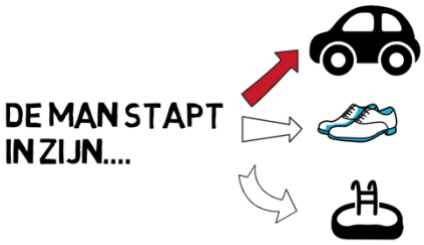
1. /`d E l f t`/
2. /`d E l @ f t`/
3. /`d E l Y f t`/

3.2.7 Klinkt als….

De beste fonetische transcripties krijg je wanneer een fonetisch expert de transcripties (incl. de verschillende uitspraakvariaties) direct in SAMPA invoert. Alleen zijn er maar weinig fonetisch experts beschikbaar. Een second best optie is om de onbekende woorden (d.w.z. de woorden die volgens de spraakherkenner NIET in de woordenlijst voorkomen) fonetisch te transcriberen via een ‘klinkt als…’ manier. Bij een woord als XS4ALL zou je kunnen aangeven dat het klinkt als ‘ekses foor al’ en deze drie woorden door de G2P-engine laten transcriberen. Door de transcripties vervolgens via een TTS-engine (Tekst-to-Speech) weer hoorbaar te maken, kun je

zelf horen of de transcriptie geslaagd is of niet. Als het niet lukt op een 'klinkt als….' manier, dan zal het woord alsnog door een fonetisch expert getranscribeerd moeten worden.

3.3 Taalmodel



De reden dat mensen spraak in hun eigen taal ook in 'beroerde omstandigheden' (zeer) goed kunnen verstaan, is dat ze, wanneer de context van het verhaal bekend is, de woorden die hoogstwaarschijnlijk zullen worden uitgesproken, goed kunnen voorspellen. Wanneer een Nederlander gevraagd wordt de zin "de man stapt in zijn….." af te maken, dan geeft (bijna) iedereen **auto** op. Voor native Nederlanders is de kans op auto blijkbaar erg hoog hoewel iemand ook 'zwembad', 'sokken', 'comfort zone', of 'vliegtuig' had kunnen zeggen.

Maar wanneer we de zin "de man stapt in zijn….." horen, dan verwachten we al dat het gevolgd wordt door 'auto'. Ook in een zin als "gisteren op het politiebu…" treedt dit effect op. Na politiebu kunnen nog maar heel weinig woorden volgen en politiebureau is veruit het meest waarschijnlijk.

Maar wanneer we dit 'voorspellen' doen voor een andere taal die we minder goed kennen, dan zal de voorspelling een stuk lastiger worden: we hebben gewoon geen goede statische representatie van de taal in ons hoofd zitten om een Italiaanse zin als 'vado a ca…' goed af te maken. We moeten dus het hele woord horen voor we de herkenning kunnen starten. In de regel kun je stellen dat hoe beter we een taal kennen, hoe beter we kunnen voorspellen welke woorden waarschijnlijk zullen volgen, hoe langzamer een taal lijkt te gaan en hoe beter onze herkenning ervan is.

Voor de automatische spraakherkenning geldt hetzelfde. Het statische model dat voorspelt wat de kans is op woord X gegeven de woorden A, B en C, heet een **taalmodel**. En hoe beter zo'n taalmodel is, hoe beter de spraakherkenning werkt.
Een taalmodel bestaat uit verschillende onderdelen: een woordenlijst en bi-, tri-, quatro- en zelfs pentagrammen. De woordenlijst bevat alle woorden (met hun fonetische transcriptie) die herkend kunnen worden. Staat een woord (bijvoorbeeld tandpasta) er niet in, dan kan het ook niet herkend worden. De bi- t/m pentagrammen, zijn combinaties van 2, 3, 4 en 5 woorden uit de woordenlijst en geven je de kans op woord X gegeven de woorden A, AB, ABC en ABCD.

Hoe meer woorden je gebruikt om woord X te voorspellen, hoe beter de herkenning. Maar ook hoe groter het taalmodel wordt en dus het beslag op het geheugen en de snelheid van de herkenning. De KALDI-herkenner gebruikt op dit moment quatrogrammen waarmee nog realtime herkend kan worden.

De hoeveelheid woorden die je kunt opnemen, is beperkt en dus moet er altijd een keuze gemaakt worden: welke woorden worden wel en welke dus niet gebruikt. Normaal gesproken zijn dit de meest gebruikte woorden omdat je mag verwachten dat die ook het vaakst gesproken zullen worden. Maar dan nog zal er een keuze gemaakt moeten worden in het bronmateriaal dat je gebruikt.

Voor de eerste LVCSR[2] op de Universiteit Twente kregen we toestemming om de kranten van de PCM-groep te gebruiken: NRC, Volkskrant, Trouw en AD. Maar de Telegraaf en allerlei regionale kranten zaten daar niet bij, net zomin als tijdschriften als Vrij Nederland, de Groene of Elsevier. De meest frequente woorden waren dus de meest frequente woorden in een subset van wat er in Nederland op papier verschijnt! Het taalmodel reflecteerde dus deze bias en was eigenlijk een 'hogeropgeleiden'-taalmodel.

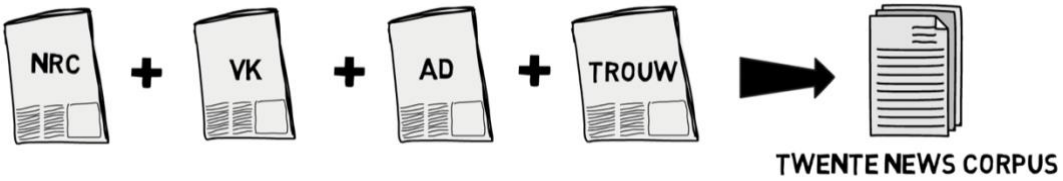


Fig. 5: Het Twente News Corpus (TNC) is gemaakt met vier verschillende kranten vanaf 1994 tot nu.

Voor het herkennen van nieuwsuitzendingen maakte dit in de praktijk niet zoveel uit maar anders wordt het voor meer specifieke spraak zoals een gesprek over de buitenlandse politiek, over de ontwikkelingen op het gebied van medicijnontwikkeling of over de identiteitspolitiek op Nederlandse Universiteiten. Dat soort gesprekken kennen een duidelijk ander woordgebruik en een standaard taalmodel zal dan veel woorden niet kunnen herkennen.

3.3.1 Specifieke taalmodellen

Een taalmodel past dus bij een type conversatie. Een gesprek over de buitenlandse politiek in de 19de eeuw zal andere woorden en woordcombinaties bevatten dan een gesprek over de ethiek van kunstmatige intelligentie. Voor de herkenning van elk van deze onderwerpen, zouden we dus een eigen taalmodel moeten maken.

3.3.1.1 Context

Een en ander werd duidelijk bij de herkenning van Oral History interviews die vooral over de Tweede Wereldoorlog gingen. Namen van concentratiekampen (Auschwitz, Treblinka), Duitse militaire begrippen (Obersturmbahnführer) en namen van voor die tijd relevante personen (Seyss-Inquart, Himmler) werden niet herkend maar er werd wel op gezocht.

Om een taalmodel te maken dat beter geschikt was voor het herkennen van OH-interviews, werden de transcripties van ± 600 interviews van het programma 'Erfgoed van de Oorlog' gebruikt om het taalmodel aan te passen.

Eenzelfde actie werd ondernomen voor het herkennen van parlementaire debatten. Hiervoor werden 3 jaar handelingen gebruikt waardoor ook woorden als **milieurapportagerapport** herkend konden worden.

3.3.1.2 Tijd

Wanneer gesprekken over specifieke onderwerpen gaan, dan zal er een context-afhankelijk taalmodel gemaakt moeten worden. Niet alleen om de herkenning te verbeteren, maar ook

[2] LVCSR: large vocabulary continuous speech recognition. Een term die 20 jaar geleden geïntroduceerd werd om het verschil aan te geven met spraakherkenners die slechts een beperkt aantal woorden konden herkennen en met spraakherkenners waarbij je een pauze moest inlassen tussen elk woord. Tegenwoordig zijn eigenlijk alle spraakherkenners LVCSR

omdat men juist op die contextspecifieke woorden zal gaan zoeken. Maar ook het standaard Nederlandse taalmodel is aan veranderingen onderhevig. Woorden en de namen van belangrijke mensen komen en gaan. Voorbeelden hiervan zijn Vuurwerkramp (sinds 2000), Facebook (sinds 2004), Twitter (sinds 2006), Hyves (2004 - 2013) en de namen van bijvoorbeeld Clinton (Bill, relevant van 1992 tot 2002 en Hillary, relevant van 2008-2016).

3.3.2 Adaptatie

Kortom, een taalmodel moet aansluiten bij de spraak die herkend moet worden en omdat die spraak tijd- en contextgevoelig is, zullen voor een optimaal resultaat contextspecifieke modellen gemaakt moeten worden die bovendien ook steeds geüpdatet zullen moeten worden.

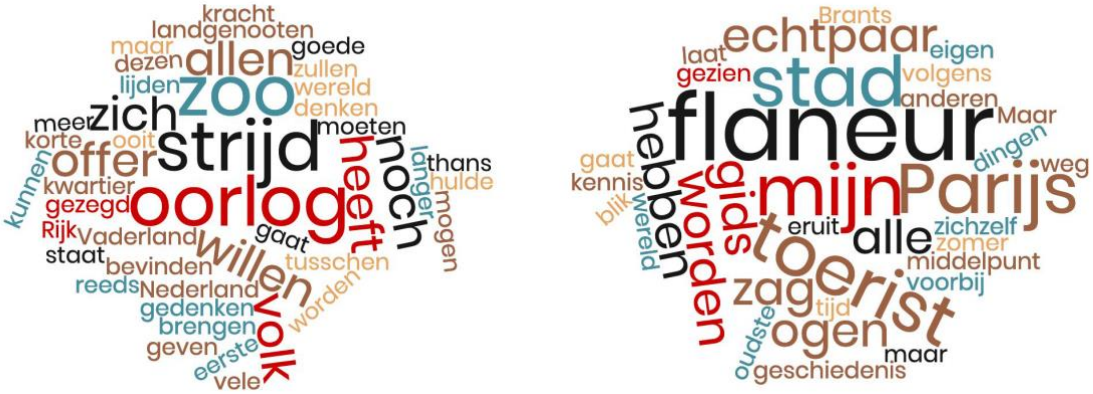


Fig. 6: Twee wordclouds, die net als een taalmodel, context afhankelijk zijn. Links gebaseerd op een toespraak van Koningin Wilhelmina voor Radio Oranje, rechts een artikel van Bas Heijne in het NRC.

4 Correctie

In het voorgaande hoofdstuk hebben we gezien dat het herkennen van spraak een veel complexere taak is dan het simpel overzetten van geluid in tekst. Mensen houden bij het spreken meestal sterk rekening met hun verwachtingen over hun gehoor en zullen woorden die door de context goed voorspelbaar zijn, “sloppy” kunnen uitspreken zonder dat de boodschap verloren gaat. Verder maken alle mensen fouten in hun uitspraak en maakt de spraakherkenner fouten bij het herkennen van de spraak.

Dit houdt in dat automatisch gegenereerde transcripten altijd gecontroleerd en meestal ook gecorrigeerd zullen moeten worden.

‘Unsupervised’ spraakherkennen zonder controle en correctie achteraf, is alleen een optie bij het ontsluiten van grote hoeveelheden spraak (distant listening). Voor verbatim transcriptie, grammaticaal correcte ‘Close Captions’ en samenvattingen/ondertitels zullen de herkenningsresultaten gecontroleerd en gecorrigeerd moeten worden. Helaas is dit een (zeer) tijdrovende klus. Een verbatim of zeer Closed Caption van scratch, kost ongeveer 8 keer de spraaktijd.

Daarom speelt het doel van de transcripties een belangrijke rol. Gaat het om een verbatim transcriptie, om een grammaticaal correcte tekst die erg dicht bij de spraak blijft of om een leesbare samenvatting van die real-time kan worden meegelezen (bijvoorbeeld ondertitels)?

Alle vormen anders dan een verbatim transcriptie zijn in wezen interpretaties van de spraak en hebben hun eigen mores. Voorop moet staan: wat is het doel van de herkenning en hoe ga je dat zo goed mogelijk met zo weinig mogelijk inspanning bereiken?

4.1 Dedicated editors

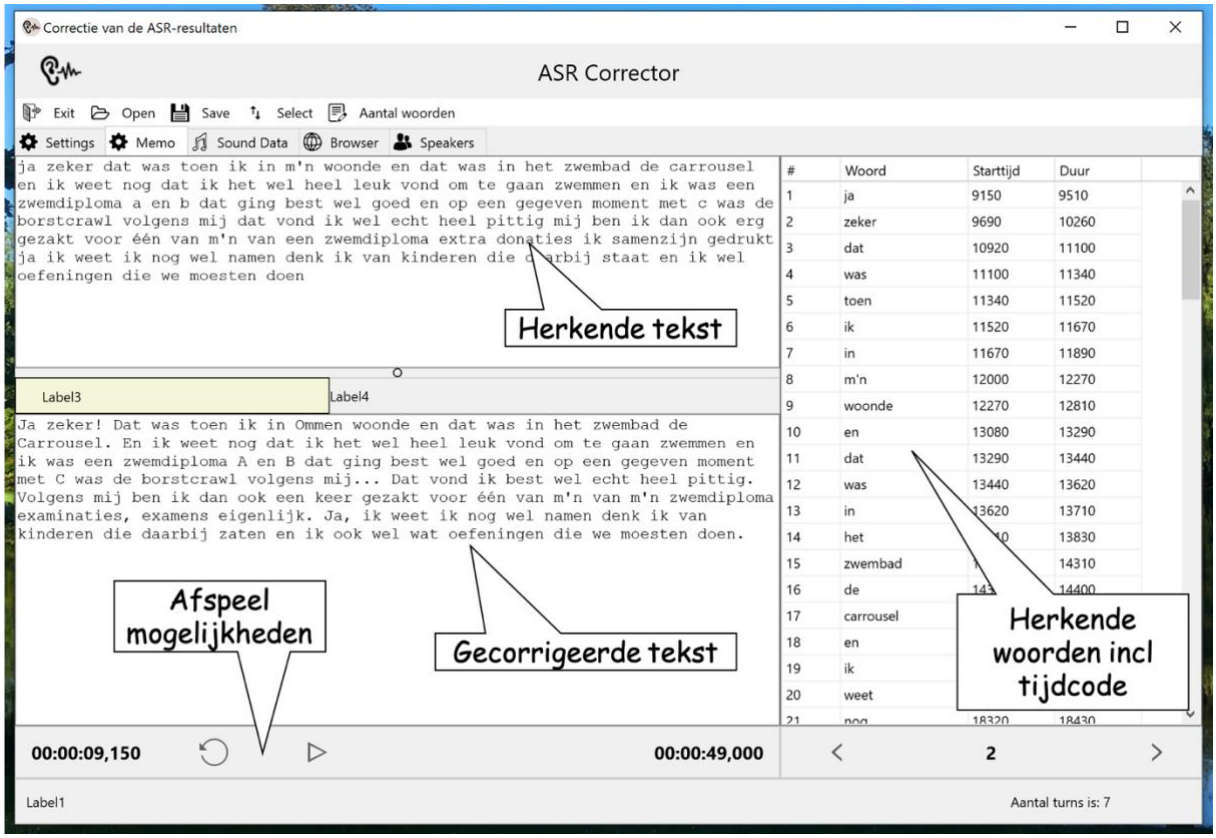


Fig. 7: Screenshot van de ASR-corrector. Een speciale tekstverwerker waarmee de herkenningsresultaten van de Nederlandse KALDI-herkenner makkelijk gecorrigeerd kunnen worden. Ieder fragment kan vanuit de editor afgespeeld worden.

In het ideale geval is er een speciale editor waarbij de relatie tussen de tekst (herkende woorden) en spraak (audiofile) behouden blijft zodat in een programma de tekst beluisterd en gecorrigeerd kan worden en waarmee de uiteindelijke (gecorrigeerde) tekst middels forced alignment opnieuw op tijd gezet kan worden.

Naast de hier getoonde ASR-corrector (in ontwikkeling) zijn er verschillende betaalde en gratis programma's beschikbaar die elk hun voor- en nadelen kennen.

Een goede gratis editor is de open source editor SubtitleEdit (https://www.nikse.dk/SubtitleEdit). Aanvankelijk bedoeld voor het ondertitelen door vrijwilligers van video's, heeft SubtitleEdit zich ontwikkeld tot een goede tool om de fouten van de spraakherkenner te corrigeren. De herkende spraak wordt als een ondertitelingsbestand weggeschreven (SRT of VTT) en ingeladen in SubtitleEdit. Doordat het programma ook de golfvorm toont kan zeer nauwkeurig gewerkt worden. Een aardige bijkomstigheid is dat de ondertitels via een koppeling met Google Translate direct in heel veel andere talen vertaald kan worden waarbij de band tussen tijd en tekst (en dus de vertaling) behouden blijft.

Het corrigeren van redelijk goed herkende teksten gaat een stuk sneller dan het handmatig transcriberen van de spraak, maar kost, afhankelijk van de granulariteit[3] van de transcriptie toch behoorlijk wat tijd.

Een aardig overzicht van andere transcription tools (in 2020) kan hier gevonden worden: https://blog.ai-media.tv/blog/best-free-transcription-tools. Sommige tools hebben de mogelijkheid om een voetpedaal te gebruiken waarmee de audio opnieuw beluisterd of gestopt kan worden zodat de handen voor het typen beschikbaar blijven.

Fig. 8: De gratis tool oTranscribe (https://otranscribe.com/) waarmee je vlot de transcripties van AV-bestanden kunt maken. De tijdcodes zijn veel minder nauwkeurig dan met SubtitleEdit, maar dat kan verholpen worden door na de transcriptie Forced Alignment te doen.

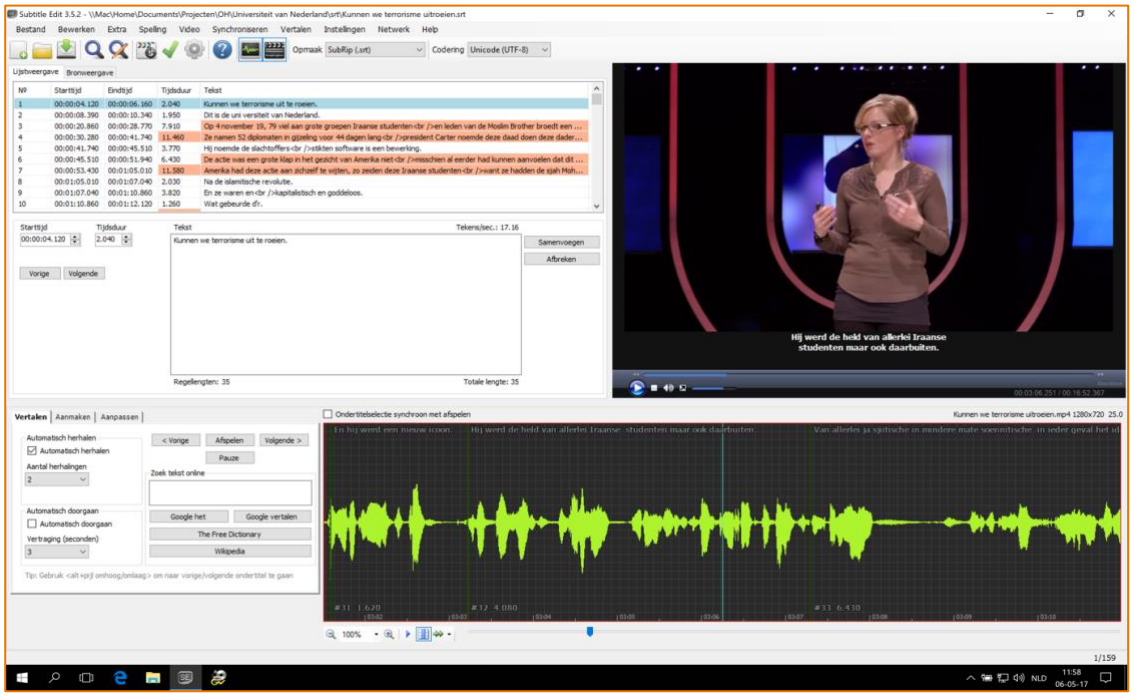


Fig. 9: Screenshot van SubtitleEdit bij het bewerken van een video van de Universiteit van Nederland. De presentatie van Beatrijs de Graaf werd eerst door de herkenner gehaald en als ondertitelingsbestand weggeschreven. Dat bestand (.srt) werd vervolgens in SubtitleEdit ingelezen en handmatig gecorrigeerd.*

4.2 Gewone tekstverwerker

Een andere manier is het kopiëren van de herkende tekst in een tekstverwerker om vervolgens de tekst te corrigeren. De relatie tussen tekst en spraak gaat hierbij verloren. Als de tekst

[3] De fijnmazigheid waarin spraak met tekst wordt opgelijnd: van grof (op paragraafniveau) tot fijn (op woordniveau) tot zeer fijn (syllabeniveau).

eenmaal in orde is, kan de gecorrigeerde tekst worden opgelijnd met de spraak (FA, Forced Alignment). Dit geeft dan een zeer nauwkeurig resultaat waarbij van ieder woord precies bekend is wanneer het werd uitgesproken. Voordeel is dat elke tekstverwerker gebruikt kan worden maar de herkenning (= FA) moet wel opnieuw gedaan worden.

Een voorbeeld hiervan is de ondertiteling van de debatten in de Tweede Kamer. Hier wordt de tekst door de medewerkers gemaakt en vervolgens worden spraak en tekst opgelijnd.

Fig. 10: Het CLARIN-D programma WebMaus is een goede OpenSource tool voor het oplijnen van de tekst en spraak. Voor een optimaal resultaat moeten er om de zoveel minuten time-stamps gezet worden zodat het oplijnen niet uit de bocht kan vliegen.

5 Presentatie ASR-resultaten

Wanneer er sprake is van ASR voor AV-materiaal, ligt het voor de hand dat er ondertitels gemaakt worden die onder de video 'geplakt' kunnen worden. Dit heeft wel tot gevolg dat het doornemen van een AV-document even lang duurt als het AV-document zelf. Het (vluchtig) lezen van de tekst gaat in de regel een stuk sneller en dus zou het een mooie toevoeging zijn als de tekst van een AV-document zelfstandig gelezen zou kunnen worden waarbij er wel een direct relatie tussen de tekst en de audio blijft bestaan (je leest de tekst en klikt daar waar je het originele document wilt bekijken/beluisteren, op de bijbehorende woorden; het AV-document begint dan vanaf dat moment te spelen). Deze manier van presenteren wordt ook wel de karaoke-presentatie genoemd.

Fig. 11: Twee verschillende manieren om de spraakherkenning te demonstreren. Links als ondertitels, rechts als een karaoke-bestand waarbij het woord dat wordt uitgesproken, wordt gehighlight.

5.1 Diarisatie

Behalve spraakherkenning is er ook **sprekerherkenning**: wie spreekt er? De techniek is goed ontwikkeld maar om het te kunnen toepassen is het nodig een database met sprekers te hebben waarmee de onbekende spreker vergeleken kan worden. Vaak is zo'n database er niet en blijft de sprekerherkenning beperkt tot sprekerdiarisatie: het aangeven van een sprekerwisseling. De software vergelijkt steeds een fragment (F_N) met het voorafgaande fragment (F_{N-1}). Is het verschil groter dan een drempelwaarde dan wordt ervan uitgegaan dat er een andere spreker is. Het fragment F_N kan vervolgens vergeleken worden met alle voorafgaande fragmenten (F_{N-2}, F_{N-3}, F_{N-4}, …. $F_{1)}$. Wanneer het fragment F_N erg lijkt op fragment F_{N-x} dan kan de software besluiten dat de sprekers van F_N en F_{N-x} dezelfde zijn.

Het onderverdelen van de herkende tekst in sprekers verhoogt zowel de leesbaarheid als de ontsluitingsmogelijkheden van de AV-opnamen (zie de twee voorbeelden hieronder)

5.2 Zin-generatie

De leesbaarheid van de tekst neemt enorm toe wanneer de herkende tekst wordt omgezet in (pseudo)zinnen. Zoals gezegd: mensen spreken in de regel niet in zinnen en dus is het lastig om de herkende spraak in grammaticaal correcte zinnen om te zetten. Toch kan er wel een poging gedaan worden door bijvoorbeeld pauzes van meer dan 400 ms als een zinseinde te beschouwen. Vaak gaat dat redelijk goed, maar bij langzame of aarzelende sprekers levert dat soms te veel 'zinnen' op. Toch is dit beter dan een grote rij van herkende woorden.

5.2.1 Alleen herkende tekst

Nieuwsradio bnr eye opener festival wordt voor het eerst op grote schaal een nieuwe tekst technologie gebruikt waardoor bezoekers van lezingen de vertaling direct op hun mobiel te zien krijgen arjan van hessen van de universiteit twente is één van de sprekers tijdens het festival welkom in de uitzending wat wat is er zo bijzonder aan deze nieuwe technologie die tijdens het festival getest gaat worden nou vooral dat dat nu eens een keer echt in de praktijk getest en wij zijn al jaren bezig met dit soort het zal zeker niet perfect maar we zien in toenemende mate dat door steeds beter dan automatisch kralingen steeds betere spraken naar de langzaam in de buurt komen om als de inhoud niet te ingewikkeld is heel of bijna vertaling gaan dat je kunt het vergelijken met die tot die simultaan vertaald wie doet dat ongetwijfeld nog beter maar we komen een heel eind in de richting dus ik ben ook heel heel erg benieuwd naar wat morgen en overmorgen op vrijdag en zaterdag

5.2.2 Herkende tekst met spreker diarisatieen pseudo-zindetectie

Wanneer sprekerdiarisatie wordt gebruikt en pauzes van 400 ms en langer als zinseinde worden beschouwd, dan kan een soort pseudo-nette tekst gemaakt worden. Het is zeker niet foutloos, maar verhoogt desondanks de leesbaarheid.

S0

Nieuwsradio. Bnr eye opener

S1

Festival wordt voor het eerst op grote schaal een nieuwe. Tekst technologie gebruikt waardoor bezoekers van lezingen de vertaling direct op hun mobiel te zien krijgen. Arjan van hessen van de universiteit twente is één van de sprekers tijdens het festival welkom in de uitzending. Wat wat is er zo bijzonder aan deze nieuwe technologie die tijdens het festival getest gaat worden. Nou

S14

Vooral dat dat nu eens een keer echt in de praktijk getest. En. Wij zijn al jaren bezig met dit soort. Het zal zeker niet perfect. Maar we. Zien in toenemende mate dat. Door steeds beter dan automatisch kralingen steeds betere. Spraken naar. De langzaam in de buurt komen om als de inhoud niet te ingewikkeld is. Heel. Of bijna. Vertaling gaan dat je kunt het vergelijken met die tot die simultaan vertaald wie doet dat ongetwijfeld nog beter maar we komen een heel eind in de richting dus ik ben ook heel heel erg benieuwd. Naar wat. Morgen en overmorgen. Op vrijdag en zaterdag.

5.3 Presenteren zoekresultaten

Het ontsluiten van de grote hoeveelheden AV-materiaal middels het automatisch transcriberen levert de mogelijkheid om er snel en makkelijk in te kunnen zoeken. Het is echter niet altijd duidelijk hoe de resultaten gepresenteerd moeten worden. Meestal worden de AV-bestanden bij het zoeken één voor één behandeld (bijvoorbeeld alfabetisch of op datum) en worden alle voorkomens van het zoekwoord inclusief de tijdcode in die volgorde teruggegeven. Door vervolgens op een zoekresultaat te klikken, wordt AV-document vanaf dat moment (het tijdstip waarop het zoekwoord volgens de herkenner gezegd werd) aan de gebruiker getoond. Een mooi voorbeeld is te zien bij het NIOD-project Getuigenverhalen (zoek bijvoorbeeld op ‘honger‘).

Fig. 12: Een screenshot van de zoekresultaten van het woord 'honger' in de 600 interviews van getuigenverhalen. De tijden geven de tijd aan van het woord honger. In de rechterkolom staat de ondertitelregel met daarin het woord honger zodat het direct in context staat. Door op de tijd te klikken, wordt het videofragment vanaf die tijd afgespeeld.

Fig. 13: Door op de tijd te klikken (fig. 12) wordt de video geopend vanaf die tijd (hier op 17:13). Op die manier kan heel snel in de video gezocht worden en kunnen de gezochte fragmenten direct beluisterd/bekeken worden.

Maar het is de vraag of een lineaire manier van presenteren (op tijd, alfabetisch) altijd de juiste is. Het presenteren van zoekresultaten is een vak op zich en valt buiten de scope van dit rapport, maar het is wel iets om goed over na te denken.

6 Automatisch vertalen

Door de snel toenemende internationalisering van het hoger onderwijs zal het materiaal niet alleen in het Nederlands maar ook in het Engels ontsloten en gebruikt moeten kunnen worden. Voor AV-materiaal dat in het Engels is opgenomen (nu zo'n 50%) gaat dit vanzelf, maar het Nederlandstalige materiaal moet middels (automatische) vertalingen ontsloten worden. Daarnaast is het zo dat lang niet alle gebruikers (wo- en hbo-studenten) het Engels zo goed machtig zijn, dat er geen behoefte is aan een goede Nederlandse vertaling. Het is dus verstandig om ervan uit te gaan dat alles in beide talen beschikbaar moet zijn.

Maar om goede automatische vertaling (AT) te kunnen gebruiken, moet de input (d.w.z. de tekst die uit de spraakherkenner komt) zo veel mogelijk uit grammaticaal correcte zinnen bestaan. Een spraakherkenner levert echter in de regel geen zinnen op maar 'slechts' een rij opeenvolgende (herkende) woorden. Wanneer de AT-software geen (correcte) zinnen krijgt aangeboden, worden de opeenvolgende woorden letterlijk vertaald. Voor het begrip van wat er gezegd werd is dit niet erg bevorderlijk. Wanneer AT zou moeten worden toegepast, dan is het noodzakelijk om de herkende tekst eerst te transformeren naar (pseudo)zinnen en eventueel de grootste fouten handmatig te corrigeren. Als dat gedaan is, dan is de AT van bijvoorbeeld https://DeepL.com voldoende goed om zeer bruikbare resultaten te geven.

DeepL probeert bovendien uit een rij opeenvolgende woorden, zelf zinnen te maken hetgeen soms best goed lukt!

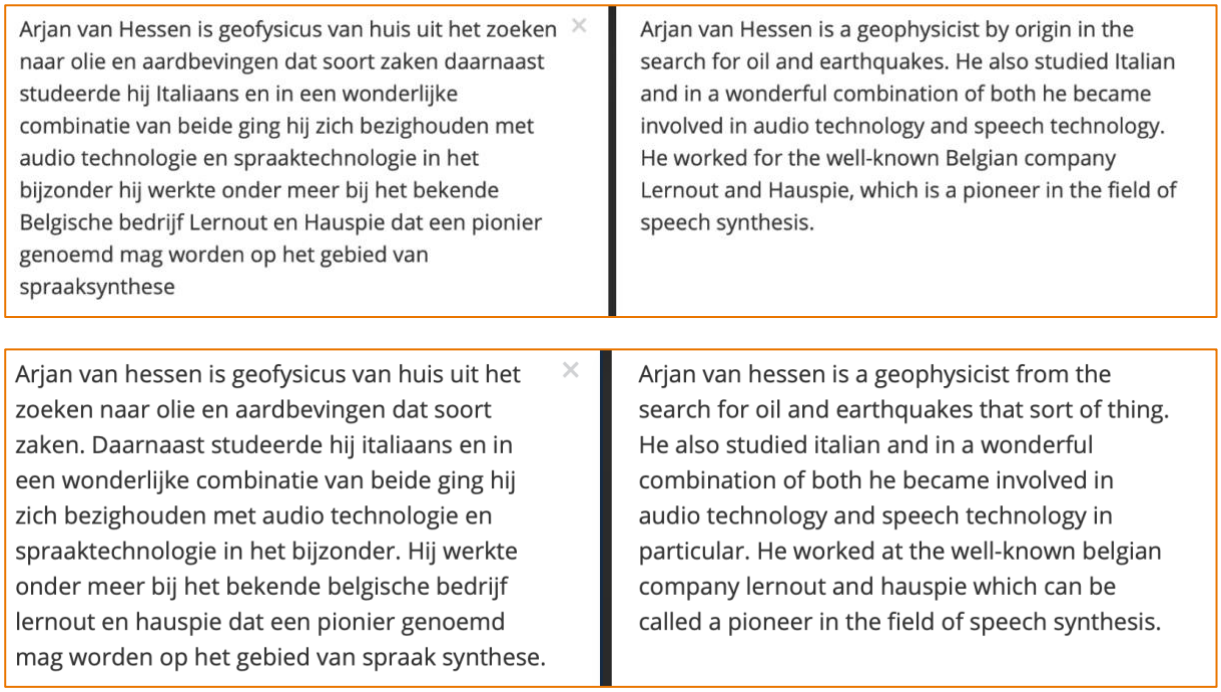


Fig. 14: Automatisch vertaling van het NL naar het EN gebaseerd op een rij (goed) herkende woorden (boven) en van dezelfde tekst waarvan zinnen gemaakt zijn (onder).

7 Automatische spraakherkenning: hoe pak je het aan?

In het voorgaande hebben we geprobeerd een overzicht (inclusief wat achtergrondinformatie) te geven over de verschillende pijlers waar automatische spraakherkenning op draait: opnamen en bronmateriaal, akoestisch model, taalmodel, grafeem-naar-foneem-transformatie, correctie en wijze van presenteren.

In het volgende gaan we dieper in op het 'hoe'. Wat moet je doen om een geschikt ASR-ecosysteem op te zetten (en te onderhouden) waarmee de verschillende video's van SURF geprocest kunnen worden.

Opnieuw zullen we naar de verschillende pijlers kijken waarbij we ons vooral zullen focussen bij alles wat komt kijken bij het taalmodel.

7.1 Akoestisch model

Een akoestisch model (AM) bestaat uit een verzameling van fonemen en diphonen (combinatie van 2 opeenvolgende fonemen) die voor bepaalde taal relevant zijn. Voor het Nederlands zijn dit er ongeveer 48 (inclusief fonemen uit andere talen die in het Nederlands gebruikt worden). Een AM wordt gemaakt door grote hoeveelheden Nederlands op te nemen, verbatim te transcriberen en vervolgens alle woorden fonetisch te transcriberen. Dit laatste kan deels automatisch met PRAAT.

Hieronder een voorbeeld van zo'n fonetisch transcriptie. Wanneer van elk woord bekend is wanneer het werd uitgesproken (3de regel van onder) dan kan vervolgens de officiële (canonische) transcriptie van elk woord op tijd gezet worden (1ste regel van boven). Dan wordt er wel van uit gegaan dat de spreker de officiële uitspraak gebruikt (dus verzekering -> `v@r-ze:-k@-rIN` en niet `f@-se:-krIN`) en niet de normale manier van spreken.

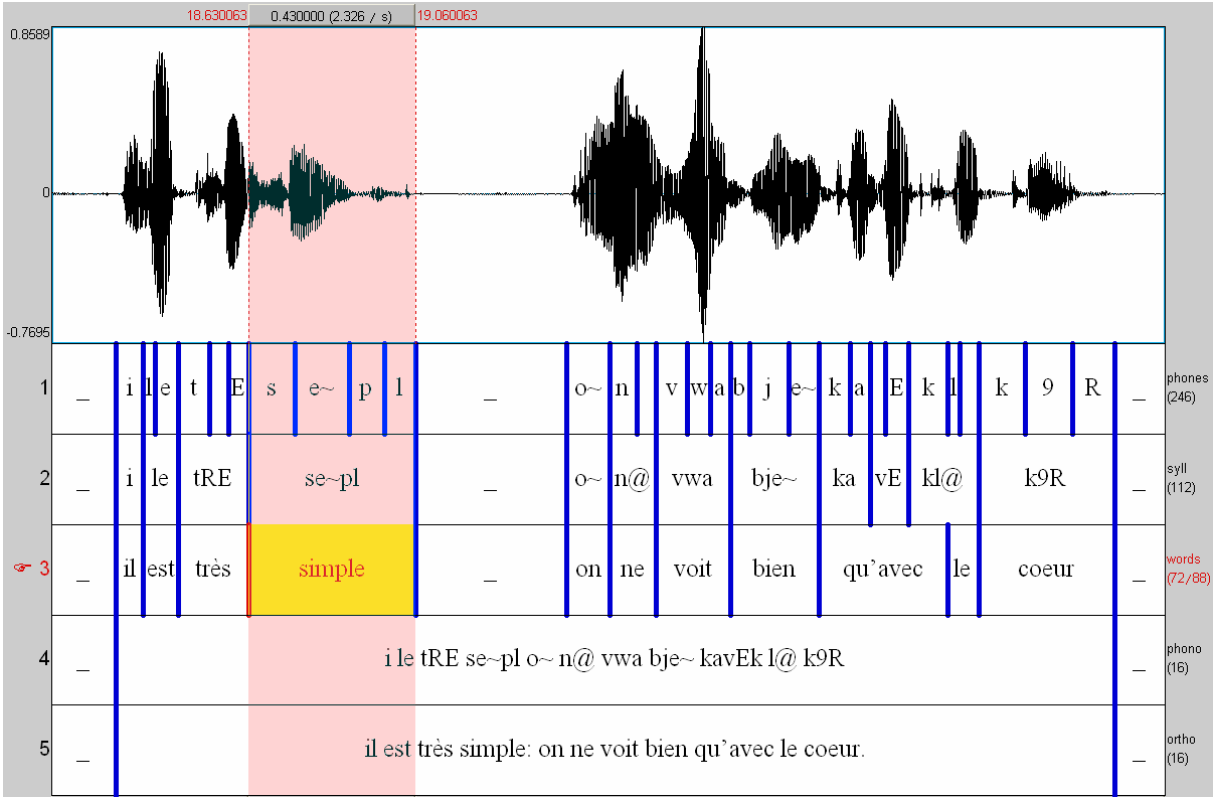


Fig. 15: Screenshot van de fonetische alignment in PRAAT. Van onder naar boven zien we de geschreven zin (regel 5), de fonetische versie daarvan (regel 4). Dan op regel 3 de alignment van de woorden van regel 5 met het moment van spreken. In regel 2 staan de opgelijnde woorden opnieuw maar nu als lettergrepen. De eerste regel tenslotte bevat de opgelijnde fonemen. Deze opgelijnde fonemen kunnen vervolgens gebruikt worden om een akoestisch model te maken of aan te passen.

Op deze manier kunnen, wanneer voldoende materiaal beschikbaar is, de gemiddelde manier van uitspreken van de Nederlandse fonemen berekend worden. Uiteraard is het niet zo dat een foneem in iedere positie hetzelfde wordt uitgesproken. Een begin /p/ klinkt anders dan een midden of eind /p/ en dus is het verstandig om niet **een** foneem te berekenen maar van elk foneem een positie-versie te berekenen.

Hoe dan ook: op bovenstaande manier wordt een akoestisch model gemaakt. Maar omdat de uitspraak van een taal kan veranderen, is het verstandig om het AM om de zoveel tijd bij te werken zodat het beter overeenkomt met de manier van spreken van het te herkennen materiaal.

Een mooi voorbeeld van het anders uitspreken van het Nederlands is het zo geheten Poldernederlands. Onder invloed van hoogopgeleide, jonge vrouwen veranderde de manier van uitspreken van bijvoorbeeld de /ij/.

Poldernederlands is de term die de sociolinguïst Jan Stroop bedacht heeft om het verschijnsel aan te duiden dat de AN-tweeklanken ei /ɛi/, ui /œy/ en ou /ɔu/ met een meer open mond (lagere kaakstand) worden uitgesproken en zo gaan neigen naar aai /æi/ of /ai/, ou /ʌy/ en aau /ɑu/ of /au/. Zo klonk *Blijf bij mij*, gezongen door Ruth Jacott alsook door Paul de Leeuw[1], als *"Blaaif baai maai"*, en zingt Marco Borsato in het liedje Rood niet alleen *'blaauw'* maar ook *'jaau'* ('jou') en *'haau'* ('hou').
Wikipedia

Fig. 16: Beschrijving en grafische weergave van het Poldernederlands (Jan Stroop): de verschuiving van de uitspraak van klinkers in de uitspraak van jonge, hoogopgeleide vrouwen.

Als dit Poldernederlands algemeen gebruikt zou gaan worden, dan zal het akoestisch model voor het moderne Nederlands dus moeten worden aangepast. Ook zal het Nederlands onder invloed van de komst van bijvoorbeeld grote groepen emigranten (Marokkanen, Turken, Polen) en de internationalisering (Internet) veranderen. T.z.t. zal het algemene AM voor het Nederlands dus moeten worden aangepast.

Iets anders is het AM voor de verschillende groepen in de samenleving. Op dit moment zijn er twee Akoestische Modellen voor het Nederlands: het Noord-Nederlandse AM en het Zuid-Nederlandse AM (zeg maar het Nederlands en het Vlaams). Maar om bijvoorbeeld sprekers uit het Noorden of Oosten beter te kunnen herkennen, zou er eigenlijk ook een Noord en Oost-Nederlands AM moeten komen.

Dit geldt eigenlijk voor alle spraak die duidelijk afwijkt van het standaard Nederlands (of Vlaams). Wanneer blijkt dat veel sprekers op de video's van SURF een 'afwijkende' manier van uitspraak hebben (bijvoorbeeld omdat er allemaal Nederlandstalige docenten Engels spreken) dan zou het wellicht goed zijn om een Neder-engels AM te maken.

Wat nodig is, is dan een grote hoeveelheid verbatim getranscribeerde opnamen.

7.2 Taalmodel

Een taalmodel is niets anders dan een statistisch model dat de kans op woord X geeft, gegeven de voorafgaande woorden ABC. Het lijkt op de autocorrectie die gebruikt wordt bij mobiele telefoons.

Fig. 17: Taalmodel: autocorrectie op de mobiele telefoon.

Een taalmodel moet in het ideale geval het woordgebruik van de spreker(s) zo goed mogelijk reflecteren. Het woordgebruik van een spreker is afhankelijk van zowel de persoon in kwestie als het onderwerp waarover gesproken wordt. Zo zal een jonge Marokkaan uit Amsterdam waarschijnlijk andere woorden gebruiken dan een jonge Oost-Groninger wanneer ze beiden over hetzelfde onderwerp praten. Evenzo zal het woordgebruik anders zijn wanneer diezelfde Marokkaan (of Oost-Groninger) over moderne muziek, of over het repareren van oude auto's spreekt.

Om dit te kunnen ondervangen, zou er voor iedere groep mensen en elk onderwerp een apart taalmodel gemaakt moeten worden; iets wat eigenlijk niet kan en ook niet echt nodig is. Hoewel het woordgebruik per onderwerp kan verschillen, is het niet zo dat voor ieder gespreksonderwerp een eigen taalmodel nodig is. Met 256.000 woorden die herkend kunnen worden, kan een voldoende robuust taalmodel gemaakt worden dat in veel gevallen de gesproken tekst goed kan herkennen. Een taalmodel moet pas worden aangepast wanneer het zulke contextspecifieke woorden bevat dat de herkenning er sterk onder lijdt. Denk bijvoorbeeld aan gesprekken over marketing, lessen voor verpleegkundigen of studenten geneeskunde, interviews over de buitenlandse politiek of interviews over het tot stand komen van een modern kunstwerk.

7.2.1 Achtergrond van een taalmodel

Een taalmodel wordt altijd gemaakt met grote hoeveelheden tekst. Van de tekst worden de opeenvolgende zinnen als het ware onder elkaar gezet en wordt vervolgens gekeken hoe groot de kans op het woord X of het woord Y is wanneer ze worden voorafgegaan door de woorden ABC. Komt in een tekst bijvoorbeeld de zin “ik drink bier” 3 keer voor en de zin “ik drink koffie” 7 keer, dan is de kans op koffie 70% gegeven de twee woorden “ik drink”.

In het ideale geval worden de kansen berekend op basis van de gesproken tekst. Maar dat is lastig en vooral duur, omdat dan enorme hoeveelheden gesproken spraak verbatim getranscribeerd moeten worden en een uur spraak ongeveer 8 uur uitwerken kost.

Daarom worden meestal geschreven teksten gebruikt, ook al weten we dat spreektaal en schrijftaal van elkaar verschillen. Schrijftaal is in de regel grammaticaal correct, er wordt niet in geaarzeld, zinnen worden afgemaakt en het woordgebruik is in de regel netjes. Spreektaal daarentegen is veel meer zoekend, aarzelend, zinnen worden halverwege afgebroken, en werkwoord tijden (tegenwoordige tijd, verledentijd, voltooid verledentijd) worden door elkaar gebruikt. Omdat we om praktische redenen geschreven teksten gebruiken voor het maken van een taalmodel, is het wel verstandig om dan teksten te gebruiken die niet al te ver af staan van de spreektaal. Een doorwrochte handeling over de filosofie van Wittgenstein leent zich dan ook niet voor het maken van een taalmodel (hoewel de woorden die erin voorkomen wel gebruikt kunnen worden voor een filosofie-taalmodel).

7.2.2 Generieke taalmodel

De eerste (academische) toepassingen van de LVCSR (Large Vocabulairy Continuous Speech Recognition) lag op het gebied van het herkennen van het 8-uurjournaal (Universiteit Twente, 2003). Om hiervoor een taalmodel te maken, werd gebruik gemaakt van ongeveer 10 jaargangen kranten (Volkskrant, NRC, Trouw en het AD). Later werden ook de getranscribeerde teksten van het CGN (Corpus Gesproken Nederlands) gebruikt. Het CGN met zo’n 9 miljoen uitgeschreven woorden, bood een unieke kans om de taalmodellen richting spreektaal te verschuiven. Dit generieke taalmodel vormt de basis van de KALDI-spraakherkenner.

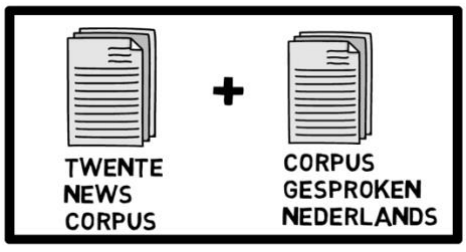


Fig. 18: Algemeen taalmodel, opgebouwd uit het Twente News Corpus en het Corpus Gesproken Nederlands.

7.2.3 Oral-history-taalmodel

Fig. 19: Oral History taalmodel door het algemene taalmodel uit te breiden met bijvoorbeeld het Koninkrijk der Nederlanden in WOII

De eerste onderzoekslijn waarin op grotere schaal SSR werd toegepast, was oral history. Veel onderzoekers in dit gebied nemen interviews af met mensen die bepaalde historische gebeurtenissen hebben meegemaakt, maar het uitschrijven van de interviews (waardoor ze makkelijker gedeeld konden worden) bleef een enorme bottleneck. Al snel werd duidelijk dat juist de voor het ontsluiten relevante woorden (Sturmbahnführer, Treblinka, Seyss-Inquart) niet in het algemene Nederlandse taalmodel stonden. Het ging hier dus niet om het veel beter maken van de herkenner, maar om het herkennen van de voor het gebruik relevante woorden. En het gebruik was vooral het zoeken en ontsluiten. Bijna niemand zoekt op het gebruik van het woordje ‘de’ maar wel op het gebruik van bijvoorbeeld ‘concentratiekamp’, ‘Trebinka’ of ‘treintransporten’. Wanneer die woorden dan niet herkend kunnen worden omdat ze niet in het taalmodel staan, dan wordt de ASR als ‘voor mij niet-werkend’ bestempeld.

Om het oral-history-taalmodel te maken, werden de transcripties van de ± 700 uur aan interviews gebruikt die in het kader van het overheidsprogramma ‘Erfgoed van de Oorlog’ gemaakt waren. Uiteraard waren de woorden van die 700 uur niet voldoende om een nieuw taalmodel te maken, maar het was wel voldoende om het bestaande Nederlandse Taalmodel te adapteren (aanpassen) waardoor de typische oral-history-woorden nu wel goed herkend kunnen worden.

7.2.4 Parlementair taalmodel

In 2012 werden onderzoekers van de Universiteit Twente (HMI-groep) gevraagd om te onderzoeken of ASR voor de Tweede Kamer voldoende goed was om zinvol te kunnen worden ingezet. Het antwoord was ja en nee.

Ja, omdat de herkenning al een stuk beter was dan tijdens een eerder project (2005) en
nee, omdat de herkenning niet goed genoeg was om automatisch ondertitels te genereren. Wat wel kon, was het oplijnen van spraak en tekst.

7.2.4.1 Forced alignment

Begonnen werd met het oplijnen van de Handelingen (door de medewerkers van de Dienst Verslaglegging en Registratie (DVR) gemaakte transcripties) met de video-opnamen. De eerste resultaten waren minder goed dan verwacht omdat de medewerkers van de DVR er te vaak ‘mooi’ Nederlands van maakten. Een gesproken zin als “ik heb dorst en daarom drink ik een biertje” werd opgeschreven als “ik drink een biertje want ik heb dorst”. Semantisch niets mis mee, maar het oplijnen gaat dan verkeerd.

Fig. 20: Voorbeeld van hoe het mis kan gaan bij het oplijnen van tekst en spraak wanneer de worden min of meer gelijk zijn maar ze in een andere volgorde staan.

Wanneer er getimed wordt op ‘drink een biertje’ dan blijft er een stukje audio over waar geen tekst bij hoort en komt er heel kort de tekst ‘ik heb dorst’ op het scherm waar geen audio bijhoort. Na verduidelijking van dit probleem besloot de DVR voortaan dichter bij gesproken tekst te blijven waardoor bovenstaande problemen verdwenen.

Nadat het oplijnen goed geregeld was, werd er gekeken naar de eigenlijke spraakherkenning. Vrij snel werd duidelijk dat, hoewel de herkenning zeker niet slecht was, juist de speciale Tweede-Kamerwoorden als bijvoorbeeld ‘milieurapportage’ niet herkend werden doordat ze niet in het taalmodel zaten. Besloten werd een dedicated Tweede-Kamer-taalmodel’ te maken van 5 jaar handelingen (iets meer dan 100 miljoen woorden).

Uiteraard wordt er normaal Nederlands gesproken in het parlement en dus kwam het nieuwe taalmodel grotendeels overeen met het nieuwe ‘parlementaire taalmodel’ maar doordat juist dat typisch politieke woordgebruik ook aanwezig was in de handelingen, werden die termen beter herkend. En nogmaals: bij het zoeken in teksten gaat het vooral om die woorden!

Bij het maken van het parlementaire taalmodel bleek dat er meer dan 759.000 verschillende woorden waren. Dat is 3 keer meer dan de herkenner aankan (dat is slechts 250.000). Dit kwam vooral door de erg lange woorden als milieurapportage en milieurapportagerapport.

7.2.5 Decompounding

De oplossing hiervoor was het decompounden (splitsen) van de woorden. Het Nederlands is, net als het Duits, een compounding language: een taal waarin nieuwe woorden gemaakt kunnen worden door ze achter elkaar te zetten. Denk hierbij aan ventiel -> ventieldopje -> ventieldopjetang -> ventieldopjetanghouder. Dat maakt dat wij meer woorden hebben dan bijvoorbeeld het Engels waar dit soort ‘aan elkaar schrijven’ veel minder vaak voorkomt.

De vraag is altijd: is de herkenning ‘fout’ wanneer ‘milieurapportagerapport’ herkend wordt als ‘milieu’+’rapportage’+’rapport’? Formeel wel, maar in de regel wordt het niet echt als fout gezien.

Maar als er gezocht wordt op ‘milieurapportagerapport’ dan wordt dat woord niet gevonden. De oplossing ligt waarschijnlijk in een vorm van post-processing waarbij de opeenvolgende losse

woorden van de herkenning weer aan elkaar geplakt worden of in het decompounden van lange zoekwoorden in hun samenstellingen om daar dan mee te gaan zoeken.

Voor het parlementaire taalmodel werden de lange, samengestelde woorden die weinig voorkwamen net zolang gesplitst tot het totaalaantal unieke woorden onder de 250.000 kwam.

7.2.6 G2P

De G2P (Grapheme-naar-Phoneme) omzetter geeft aan hoe een geschreven woord kan worden uitgesproken. De meeste G2P's combineren een 'Dictionary-Look-Up' met een set regels voor de woorden die in het woordenboek ontbreken. Ook worden in toenemende mate machine-learning-technieken gebruikt. De bekendste is die waarbij het algoritme leert wat het foneem is met 3 karakters rechts en 3 karakters links. Denk daarbij bijvoorbeeld aan het woord wortelbrood. De **e** wordt uitgesproken als een /@/ (schwa). Het venster ziet er dan uit als: **ort-e-lbr**. In normaal Nederlands: een e met de 3 letters ort ervoor en de 3 letters lbr erna, wordt uitgesproken als een /@/. Dit soort modellen worden getraind op bestaande uitspraakwoordenboeken en werken in de regel uitstekend. Maar omdat er altijd uitzonderingen zullen zijn, is het toch verstandig om ook een uitzonderingswoordenboek te gebruiken. Staat het woord daarin dan wordt die uitspraak gebruikt, zo niet dan worden of de uitspraakregels gevolgd of de machine-learning-aanpak toegepast.

Meestal gaat het goed maar het gaat fout wanneer er geïnterpreteerd moet worden. Nummers en in mindere mate afkortingen, worden context afhankelijk uitgesproken. Denk bijvoorbeeld aan het nummer 2731742 in een gewone tekstregel.

7.2.6.1 Nummers

Mensen spreken nummers/getallen uit op zo'n manier dat de andere kant ze zo goed mogelijk begrijpt. Ze gebruiken daarvoor een set contextafhankelijke regels. Zo zal niemand zijn salaris als een reeks afzonderlijke cijfers voorlezen, maar dat soms wel doen wanneer het om een telefoonnummer gaat. De regels die mensen hanteren zijn niet officieel en verschillende groepen doen dat soms op een eigen manier. Amsterdammers spreken hun postcodes zowel als 10 25 en als 1025 uit, maar in Utrecht zal bijna niemand drieduizend-vijfhonderd-vijf-en-tachtig zeggen: dat wordt 35 85.

Het voorspellen van de 'juiste manier' waarop getallen zullen worden voorgelezen blijft lastig, maar er zit wel een zekere regelmaat in de manier waarop de meeste mensen dat doen. Het is daarom verstandig om deze manier van uitspraak te gebruiken in het omzetten van getallen in een waarschijnlijke uitspraak zoals hieronder wordt getoond.

- Mijn telefoonnummer is 2731742. → zeven-en-twintig een-en-dertig zeven twee-en-veertig
- Mijn salaris is 2731742. → twee miljoen, zevenhonderd-een-en-dertigduizend zevenhonderd-twee-en-veertig
- Mijn bankrekening is 2731742. → twee -zeven-drie-een-zeven-vier-twee

Mensen hebben in de regel geen probleem met voorlezen van nummers omdat ze de context begrijpen en de uitspraak daaraan aanpassen. G2P-algoritmes doen dat niet en dus moet er een pre-parsing plaatsvinden om de nummers in de juiste context te plaatsen om ze vervolgens correct uit te kunnen spreken.

7.2.6.2 Afkortingen

Ook afkortingen volgen in de regel niet de normale manier van uitspreken. Afkortingen die geen klinkers bevatten of een combinatie van letters kennen die niet-Nederlands zijn, worden gespeld

- NTS → /E n - t e: - Es/,
- VPRO → /ve: - p e: - E r - o:/).

Afkortingen die wel uitspreekbaar zijn als gewoon woord, worden soms wel en soms niet als woord uitgesproken. Zo wordt VARA gewoon als /v a: - r a:/ uitgesproken maar het even uitspreekbare KRO wordt gespeld en niet als /k r o:/ uitgesproken.

De moderne trend om getallen in (merk)namen te verwerken zorgt ook voor verwarring. Een woord als XS4ALL heeft een achterliggend idee: toegang voor iedereen maar dan in het Engels en opgeschreven op een 'hippe' manier. Regels en machine learning gaan hier stuk op en dus zullen dit soort woorden in het uitzonderingswoordenboek moeten worden gezet.

7.2.7 Bouwen van een taalmodel

Om een taalmodel te bouwen, zijn grote hoeveelheden teksten nodig. Wanneer het om een taalmodel gaat voor sportberichten, dan is het verstandig om 'sportteksten' uit kranten, tijdschriften en websites te gebruiken. De kans dat het woordgebruik in die bronnen overeenkomt met de woorden die in sportgerelateerde spraak gesproken worden, is nu eenmaal groter.

7.2.7.1 Context gerelateerde teksten

Bij de keuze van de teksten is het dus goed om rekening te houden met het gebruik van het taalmodel. Nu moet het gebruik van contextgerelateerde woorden ook niet overdreven worden. Het is en blijft Nederlands en dus zullen de meeste woorden gewone Nederlandse woorden zijn. Bij het selecteren van de teksten kunnen dus ook andere, algemene teksten gebruikt worden. Wat voorkomen moet worden is het gebruik van specifieke teksten uit een andere context. Het heeft dus geen zin om teksten die gaan over de wereldpolitiek, Iran, Irak, Jihad te gebruiken voor een taalmodel dat voor de herkenning van voetbalverslagen gebruikt gaat worden. Los van het feit dat de herkenning er niet beter van zal worden, geeft het ook een raar beeld wanneer in de transcriptie van een voetbalwedstrijd woorden als Jihad en Afghanistan voorkomen. Zo stond in de transcriptie van een verhaal over het verblijf in een concentratiekamp de naam Cruyff. Het is niet alleen fout (wat niet vreemd is) maar wekt direct de indruk dat 'het niet werkt'.

7.2.7.2 Codering

Steeds meer teksten worden gecodeerd met UTF-8. Maar oudere teksten willen nog wel eens in plain ASCII of een andere codec staan. Voordat de teksten geprocest kunnen worden, moeten ze dus allemaal op dezelfde manier gecodeerd zijn. Bij het omschrijven van de ene naar de andere codec kan er wel eens wat misgaan.

7.2.7.3 Verschillende schrijfwijze

Iets anders waar rekening mee moet worden gehouden is de manier van schrijven. De ü die in het Duits (en dus ook in Duitse namen) veel voorkomt, wordt regelmatig geschreven als **ue** (Günther → Guenther of Gunther). Nu kan een goede G2P best met deze verschillende schrijfwijze omgaan maar het maakt het taalmodel zwakker doordat er nu drie identieke woorden in voorkomen die op verschillende manieren geschreven worden. Bij het berekenen van de kans op de gesproken Günther, moet het taal model kiezen tussen Günther (komt 60 keer

voor), Guenther (komt 30 keer voor) en Gunther (komt 10 keer voor). Als al deze versies bij elkaar geveegd zouden worden (wat terecht zou zijn) dan krijgen we een 'belangrijk' woord Günther dat 100 keer voor komt. Bovendien kan het zo zijn dat, als het minimaal aantal keren dat een woord voorkomt 15 is, Gunther 'eruit valt'. Het taalmodel bevat dan 60 keer Günther en 30 keer Guenther terwijl het eigenlijk 100 keer Günther zou moeten zijn.

Om dit soort problemen te omzeilen moeten er dus herschrijfregels gemaakt worden die (in dit geval) de verschillende schrijfwijze van Günther uniformeren.

Maar dit levert ook weer een probleem op. De naam Jansen bijvoorbeeld kan op verschillende manieren geschreven worden: Janse, Jansen, Jansens, Janson, Jansons, Jansse, Janssens, Jansze, Janze, Janzen en Janzon. Als we al deze schrijfwijze terugbrengen naar Jansen (de dominante versie) dan geeft dat weer problemen bij het herkennen van de naam Janssen als het interview over Janssen gaat.

Dit probleem is niet via het taalmodel op te lossen, maar moet via een post-processingsstap gedaan worden (bijvoorbeeld als je weet dat het over Janssen gaat, alle Jansen vervangen door Janssen).

7.2.7.4 Leestekens en hoofdletters

Bij het bouwen van een taalmodel worden eerst de zinnen in de teksten bepaald en vervolgens onder elkaar gezet.

Voorbeeld (NRC 23-1-2020)

De neef, tante en oom van Chen stonden donderdag rond lunchtijd voor de deur van haar ouderlijk huis. Normaal gesproken rent ze dan naar beneden, groet ze hen, eten ze wat. Dit keer zwaaide Chen (26) vanachter het raam naar buiten en bleef ze in haar kamer zitten. Donderdagavond zit ze daar nog. „Ik dacht: wat als ik het virus ineens blijk te hebben?", vertelt ze telefonisch. „Ik wil mijn familie beschermen." Chen reisde drie dagen geleden, toen kon dat nog, vanuit haar studiestad Wuhan naar haar ouders op het platteland, 250 kilometer verderop.

Dit wordt dan

De neef, tante en oom van Chen stonden donderdag rond lunchtijd voor de deur van haar ouderlijk huis. Normaal gesproken rent ze dan naar beneden, groet ze hen, eten ze wat. Dit keer zwaaide Chen (26) vanachter het raam naar buiten en bleef ze in haar kamer zitten. Donderdagavond zit ze daar nog. „Ik dacht: wat als ik het virus ineens blijk te hebben?", vertelt ze telefonisch. „Ik wil mijn familie beschermen." Chen reisde drie dagen geleden, toen kon dat nog, vanuit haar studiestad Wuhan naar haar ouders op het platteland, 250 kilometer verderop.

Dan worden alle leestekens en alles dat tussen haakjes staat verwijderd

De neef tante en oom van Chen stonden donderdag rond lunchtijd voor de deur van haar ouderlijk huis

Normaal gesproken rent ze dan naar beneden groet ze hen eten ze wat

Dit keer zwaaide Chen vanachter het raam naar buiten en bleef ze in haar kamer zitten

Donderdagavond zit ze daar nog

Ik dacht wat als ik het virus ineens blijk te hebben vertelt ze telefonisch

Ik wil mijn familie beschermen

Chen reisde drie dagen geleden toen kon dat nog vanuit haar studiestad Wuhan naar haar ouders op het platteland 250 kilometer verderop

Dan worden de woorden aan het begin van de zin die met een hoofdletter beginnen vervangen door een versie zonder hoofdletter, tenzij het woord **met** een hoofdletter vaker voorkomt dan het woord **zonder** hoofdletter. Om dit te kunnen doen, heb je dus woordenlijsten per taal nodig. Zo komt het woord arjan (bijna) nooit voor en is het altijd Arjan; maar bij Kok/kok is dit al lastiger.

We krijgen dan

de neef tante en oom van Chen stonden donderdag rond lunchtijd voor de deur van haar ouderlijk huis

normaal gesproken rent ze dan naar beneden groet ze hen eten ze wat

dit keer zwaaide Chen vanachter het raam naar buiten en bleef ze in haar kamer zitten

donderdagavond zit ze daar nog

ik dacht wat als ik het virus ineens blijk te hebben vertelt ze telefonisch

ik wil mijn familie beschermen

Chen reisde drie dagen geleden toen kon dat nog vanuit haar studiestad Wuhan naar haar ouders op het platteland 250 kilometer verderop

Vervolgens worden de getallen uitgeschreven

de neef tante en oom van Chen stonden donderdag rond lunchtijd voor de deur van haar ouderlijk huis

normaal gesproken rent ze dan naar beneden groet ze hen eten ze wat

dit keer zwaaide Chen vanachter het raam naar buiten en bleef ze in haar kamer zitten

donderdagavond zit ze daar nog

ik dacht wat als ik het virus ineens blijk te hebben vertelt ze telefonisch

ik wil mijn familie beschermen

Chen reisde drie dagen geleden toen kon dat nog vanuit haar studiestad Wuhan naar haar ouders op het platteland **tweehonderd vijftig** kilometer verderop

Dit is nu de eerste schone versie van de tekst voor het taalmodel

7.2.7.5 Woordenlijst met frequentie

De volgende stap is het maken van een 'unieke woordenlijst' We doen dat door alle unieke woorden onder elkaar te zetten en te tellen hoe vaak ze voorkomen. Voor het voorbeeld van hierboven wordt dat:

Woord	**#**	dan	1	lunchtijd	1
ze	6	dat	1	mijn	1
haar	4	deur	1	neef	1
Chen	3	dit	1	normaal	1
het	3	donderdag	1	oom	1
ik	3	donderdagavond	1	op	1
naar	3	drie	1	ouderlijk	1
de	2	eten	1	ouders	1
en	2	familie	1	platteland	1
nog	2	geleden	1	raam	1
van	2	gesproken	1	reisde	1
wat	2	groet	1	rent	1
als	1	hebben	1	rond	1
beneden	1	hen	1	stonden	1
beschermen	1	huis	1	studiestad	1
bleef	1	in	1	tante	1
blijk	1	ineens	1	te	1
buiten	1	kamer	1	telefonisch	1
daar	1	keer	1	toen	1
dacht	1	kilometer	1	tweehonderd	1
dagen	1	kon	1	vanachter	1

vanuit	1	virus	1	zit	1
verderop	1	voor	1	zitten	1
vertelt	1	wil	1	zwaaide	1
vijftig	1	Wuhan	1		

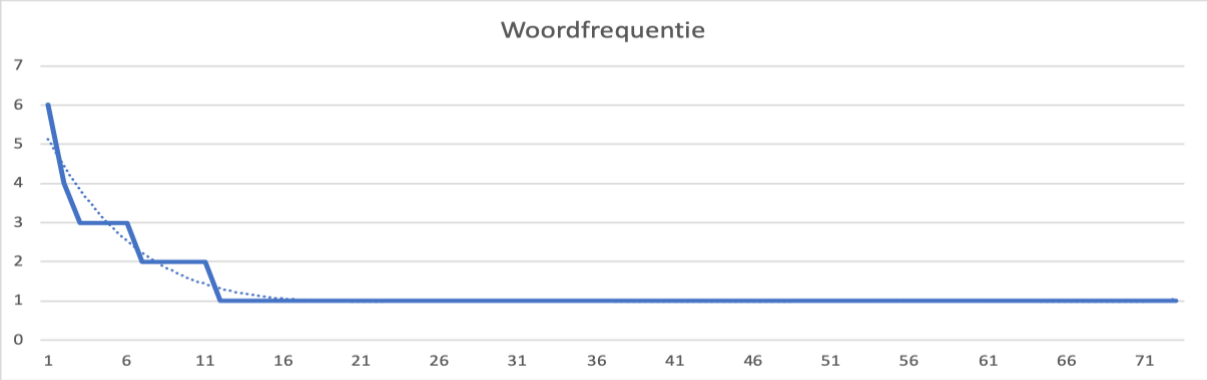


Fig. 21: Weergave van de woordfrequentie. De trendlijn (stippeltjes) geeft het normale patroon dat je ziet wanneer heel veel tekst gebruikt wordt.

Het zijn hier 94 woorden (totaal) en 73 unieke woorden. ‘Chen’ en ‘Wuhan’ komen alleen maar met een hoofdletter voor en zijn dus typische namen van een entiteit (persoon, organisatie, locatie).

7.2.7.6 N-grammen

Op basis van de schone regels worden nu de N-grammen gemaakt: de kans op woord X gegeven AB (trigrammen), geven ABC (quatrogrammen), gegeven ABCD (pentagrammen), etc. (Dit wordt door de software gedaan).

De N-grammen worden berekend op basis van de woorden die mee mogen doen. Stel dat onze herkenner 2^5 (=32) woorden kan herkennen, dan moeten de 32 frequentste woorden gebruikt worden. In de lijst hierboven is dat t/m ‘groet’. Alle woorden daaronder doen dan niet mee (de woorden die niet meegenomen worden zijn alfabetisch gesorteerd en men beargumenteren dat dat verkeerd is. In de werkelijkheid met hele grote aantallen woorden zal het niet vaak voorkomen dat woorden precies even vaak voorkomen.

7.2.7.7 Decompounden

Decompounden is altijd een beetje tricky: welke woorden wel en welke niet? Een samenstelling als voetbalcoachvergadering kan best in voetbal+coach+vergadering worden gesplitst, maar wat doe je met voetbal; voet+bal?

Er is geen perfecte oplossing, maar een goede vuistregel is dat samenstellingen die ‘vaak genoeg’ voorkomen, niet gesplitst hoeven te worden. In het voorbeeld hierboven kan besloten worden om donderdagavond alsnog te splitsen in donderdag+avond. Het woord donderdag komt dan 2 keer voor en verschuift dan in de rangorde. Het woord avond komt (door de alfabetische

volgorde) ook in de lijst en het woord donderdagavond verdwijnt. Het totaalaantal woorden wordt nu 94 – 1 (donderdagavond) + 2 (donderdag + avond) = 95.

Omdat donderdagavond nu gesplitst is, moeten in de teksten alle donderdagavond's herschreven worden in 'donderdag avond' of moeten de N-grammen waar donderdagavond in voorkomt, opnieuw berekend worden.

Wanneer alle woorden die voor decompounding in aanmerking komen, gesplitst zijn, hebben we de tweede versie van de tekst die voor de tweede versie van het taalmodel gebruikt kan worden.

7.2.7.8 Uniformiteit

Redelijk wat woorden in het Nederlands kennen verschillende schrijfwijze. Vaak zijn het van oorsprong Franse woorden die 'vernederlandst' zijn (nivo & niveau, kado & cadeau) of woorden die in de loop van de tijd (o.a. door spellingshervormingen) op een andere manier geschreven worden (reactie & reaktie, pannekoek & pannenkoek, mais & maïs). Voor de uniformiteit van de spraakherkenning en om het taalmodel zo robuust mogelijk te maken, is het verstandig om één spellingswijze te hanteren.

Fig. 22: Groene Boekje.

Een beproefde truc is om gewoon te kijken naar de variant die het meest voorkomt en die te gebruiken. Dus als mais vaker voorkomt dan maïs, dan moeten alle maïs herschreven worden. Een andere optie is om een lijst met officiële schrijfwijze aan te leggen, en die consequent door te voeren.

Naast de verschillende, goedgekeurde schrijfwijze, komen er natuurlijk enorm veel typo's voor. Zeker wanneer teksten van internet gebruikt worden maar ook in het 'nette' NRC-Handelsblad vind je doood met 3 o's. Het opsporen van deze fouten is lastig, maar kan gedaan worden door bijvoorbeeld de officiële woordenlijst van de Nederlandse Taalunie te gebruiken.

Doel van deze exercitie is het aantal verschillende woorden zo klein mogelijk te maken en de robuustheid van het taalmodel juist zo groot mogelijk. Dat dit nooit 100% zal lukken, is duidelijk en ook niet zo erg. Maar alle beetjes helpen en het verbetert de uitkomst van de herkenner.

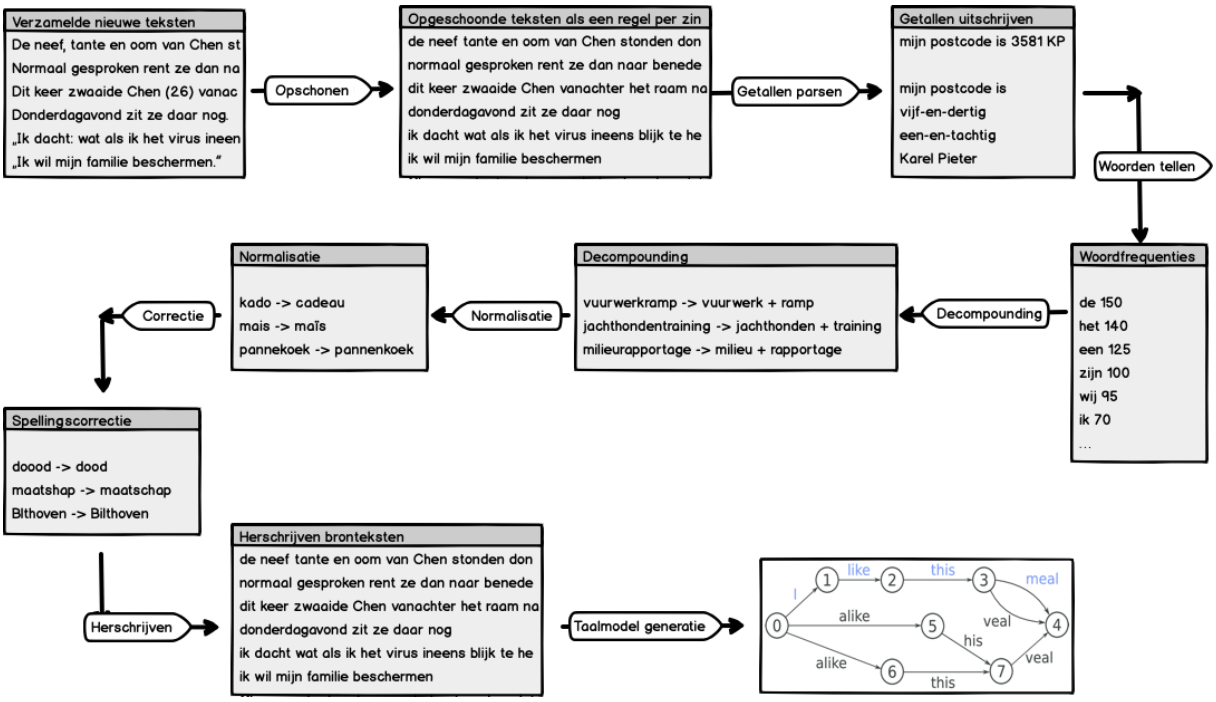


Fig. 23: Schematische weergave van de stappen die je moet nemen om van 'ruwe tekst' tot een taalmodel te komen.

7.2.8 Aantal taalmodellen

Een terugkerende vraag is: is voor het herkennen van deze spraak een apart taalmodel nodig? Het antwoord is meestal nee. De meeste spraak bevat 'gewoon' Nederlands en een gedegen Nederlands Taalmodel zal die spraak dus goed kunnen verwerken. Wanneer een gesprek over bepaalde onderwerpen gaat, dan zal het leeuwendeel van de spraak nog steeds gewoon Nederlands zijn en zullen alleen de specifieke woorden afwijken van het standaard Nederlands. Denk hierbij aan een interview over een trainingskamp van de hockeyclub. De namen van de andere clubs, typische hockeytermen en namen van de deelnemers zullen waarschijnlijk niet in de woordenlijst (en dus niet in het taalmodel) voorkomen, maar het leeuwendeel van de gesproken woorden is waarschijnlijk 'gewoon' Nederlands. Het is dan niet nodig om een nieuw taalmodel te maken, maar wel kan het taalmodel geadapteerd worden zodat de typische hockeytermen dan wel herkend kunnen worden. Zijn er nu heel veel opnamen over 'hockey gerelateerde zaken' en ook over rugby, voetbal en tennis dan is het wellicht verstandig om een sport-taalmodel te maken. Iets dergelijks werd gedaan voor oral history en het Nederlandse Parlement.

Een eigen taalmodel is dus bedoeld om een wat grotere categorie onderwerpen te behandelen. Voor het gebruik van specifieke onderwerpen binnen zo'n grotere categorie, kan het bestaande model geadapteerd worden.

7.2.8.1 SURF-modellen

De onderwerpen die in het AV-materiaal van SURF besproken worden zijn waarschijnlijk specifiek genoeg om er een eigen taalmodel voor te maken. Het zijn dikwijls academische praatjes (colleges, weblectures) met een groot aantal moeilijke en vakspecifieke woorden. Wellicht kan het standaard Nederlandse taalmodel geadapteerd worden maar waarschijnlijk is

het verstandig een eigen model te maken en dat eventueel later voor specifieke onderwerpen binnen SURF nog te adapteren (bijvoorbeeld voor colleges over medische zaken).

7.3 Maken en aanpassen taalmodel

Om een taalmodel van scratch te maken en/of om een bestaand taalmodel aan een andere context aan te passen, moeten de volgende stappen doorlopen worden.

Zie ook: https://medium.com/analytics-vidhya/a-comprehensive-guide-to-build-your-own-language-model-in-python-5141b3917d6d

7.3.1 Maken taalmodel

7.3.1.1 Verzamelen ruwe tekst

De eerste stap die genomen moet worden is het bijeenbrengen van voldoende teksten die zo goed mogelijk het woordgebruik representeren van de te herkennen spraak. De verzamelde teksten zijn in de regel geschreven en niet gesproken teksten, maar ze zullen als het goed is de voor de spraak belangrijke woorden in context bevatten.

De ruwe tekst zal afkomstig zijn uit verschillende bronnen, in verschillende formaten en uit verschillende tijden (bijvoorbeeld voor en na de spellingscorrectie). De teksten moeten eerst allemaal in platte tekst, UTF-8 herschreven worden.

In de regel geldt hoe meer hoe beter, maar een mooie hoeveelheid tekst om een robuust taalmodel te kunnen maken is ong. 10 miljoen woorden.

7.3.1.2 Herschrijven naar zinnen

De verzamelde teksten moeten in regels (lines: afgesloten met een harde return) herschreven worden waarbij iedere zin (van hoofdletter aan het begin tot punt, uitroepteken of vraagteken op het eind) op een nieuwe regel komt. Vervolgens worden alle leesteken en de teksten tussen gedachtestreepjes of haakjes, verwijderd.

7.3.1.3 Parsen van getallen

Teksten zullen in de regel ook getallen bevatten. Die getallen zullen door mensen contextafhankelijk worden uitgesproken. Dat wil zeggen: een bankrekening zal anders worden uitgesproken dan de koopprijs van een huis. De getallen in de tekst zullen dus herschreven moeten worden naar een vorm die zo goed mogelijk lijkt op wat een mens zou doen.

7.3.1.4 Bouwen woordfrequentielijst

Van de verzameling teksten worden de woorden (totaalaantal) en de unieke woorden geteld. Vervolgens wordt van ieder uniek woord bepaald hoe vaak het voorkomt (woordfrequentie). Het tellen van de woorden gebeurd case sensitief. Dat wil zeggen dat **Kok** en **kok** twee verschillende woorden zijn. Vervolgens wordt de versie die het vaakst voorkomt gebruikt en worden de ‘verliezers’ herschreven. Bij het bepalen of een woord met of zonder hoofdletter geschreven wordt, moet rekening gehouden worden met het eerste woord van de zin.

7.3.1.5 Decompounden

Om zo veel mogelijk verschillende woorden te kunnen herkennen, kan het verstandig zijn de samenstellingen (compounds) die niet zo vaak voorkomen, te splitsen (decompounden).

Wanneer voetbalschoen 10 keer voorkomt, voetbal 12 keer en schoen 40 keer, dan wordt dat na decompounden voetbal 22 keer en schoen 50 keer. Het relatieve belang van voetbal en schoen stijgt dan iets maar voetbalschoen kan niet meer herkend worden maar voetbal+schoen wel. Maar er zit een grens aan het decompounden want voetbal wordt (meestal) niet gesplitst. De ondergrens (het minimaal aantal keren dat een samenstelling kan voorkomen zonder dat het gesplitst wordt) is arbitraire en kan met trial-and-error worden gezet.

7.3.1.6 Normalisatie

Door spellingshervormingen (pannekoek ipv pannenkoek) en doordat men vaak niet weet hoe umlauten of accent circonflexe geschreven moeten worden, staan in de meeste teksten net-niet-goede versies van de woorden (b.v. mais ipv maïs). Daarnaast staan in alle teksten typo's (Wasssenaar). Het herstellen van deze woorden is verstandig om het taalmodel robuust te krijgen, maar is in de regel ook erg tijdsintensief.

Net als met de hoofd-/kleine-letter woorden moet ook nu de gehele tekst herschreven worden.

De basisgedachte bij het maken van een taalmodel is: maak het model zo robuust mogelijk opdat het (statistische) model zo goed mogelijk voorspelt welke woorden gesproken gaan worden en houdt daarbij GEEN rekening met de verschillende schrijfwijzen van woorden die hetzelfde klinken! Het aanpassen van de schrijfwijze van elk woord is iets dat in een post-processingstap gedaan moet worden (bijvoorbeeld vervang Jansen door Janssen).

7.3.2 Aanpassen Taalmodel

Het maken van een taalmodel is een behoorlijke hoeveelheid werk: verzamelen en schoonmaken van de teksten, woordfrequenties berekenen, decompounden en bepalen van de uitspraak.

Wanneer er eenmaal een goed taalmodel is, kan soms volstaan worden met het adapteren: het aanpassen van het model aan een speciale context. Wanneer de spraak in een set interviews eigenlijk gewoon Nederlands is maar er wel een aantal, voor de context belangrijke maar in het standaard Nederlands weinig voorkomende woorden gebruikt worden, dan is het makkelijker om alleen de voor die gesprekken relevante set woorden aan toe te voegen (en dus een even lange rij weinig gebruikte andere woorden te verwijderen). Dit zou kunnen door gebruik te maken van beschikbare vakvocabulaire. Maar, de herkenner kan de woorden dan wel herkennen, maar heeft niet 'geleerd' in welke woordcombinaties ze voorkomen.

Neem nu als voorbeeld het woord 'klaring' (een typisch farmaceutische term). Klaring zal veel voorkomen in zinnen als: "Om de klaring van creatinine te berekenen".

Een taalmodel 'weet' dat het niet zal voorkomen in 'ik eet klaring' maar wanneer we klaring slechts als woord toevoegen, dan zal die informatie ontbreken. Toegevoegde woorden krijgen in eerste instantie dezelfde waarschijnlijkheid als een **<unk>** (een unknown woord). Maar, het kan nu wel herkend worden! Om het relatieve belang van de nieuwe woorden te benadrukken, kunnen de nieuwe woorden met een extra gewicht (bijvoorbeeld 1000) worden toegevoegd.

Wel moeten de toegevoegde woorden dezelfde behandeling ondergaan als de nieuwe woorden bij het maken van een taalmodel.

- Verdubbelingen (creatine, kreatine, creatiene) moeten herschreven worden tot een versie (creatine)

- De uitspraak van ieder nieuw woord moet gecontroleerd worden (creatine -> kree-aa-tiene -> `k r e - a - t i: - n @`)

7.3.3 Onderhouden taalmodel

Een taalmodel is eigenlijk nooit af omdat simpel gezegd de taal verandert. Bepaalde woorden zullen minder vaak of helemaal nooit meer gebruikt worden terwijl andere woorden juist populairder worden en er nieuwe woorden bijkomen. Om 'bij de tijd' te blijven, moet een taalmodel daarom om de zoveel tijd worden aangepast. Nu verandert de taal niet zo snel dat een taalmodel maandelijks moet worden aangepast, maar het is niet verkeerd om iedere paar jaar te kijken of het model nog up-to-date is.

7.4 Resources voor de taalmodellen

Zoals hierboven geschetst, zijn er steeds verschillende stappen te nemen voor het maken of aanpassen van een taalmodel. Voor een aantal stappen is (open-source)software beschikbaar.

7.4.1 Stap 1: verzamelen geschikte teksten

Hiervoor is geen 'dedicated' software beschikbaar. Normaal gesproken worden de teksten verzameld uit bestaande documenten, woordenlijsten, transcripties enzovoort. Alle teksten moeten naar plain tekst en UTF-8 worden getranscodeerd hetgeen met standaardsoftware kan.

Een mooie collectie is het Corpus Gesproken Nederlands (CGN). Het is een collectie van 900 uur getranscribeerde spraak (9 miljoen woorden). Bijna alle Nederlandse spraakherkenners zijn gebaseerd op het CGN. Daarnaast is er een enorm tekstcorpus: Twente Nieuws Corpus met 10 jaar kranten (VK, AD, Trouw, NRC).

7.4.2 Stap 2: cleanen van de tekst

Voor het cleanen van de teksten (uniforme schrijfwijze van woorden die hetzelfde klinken, uitschrijven van getallen naar de meest waarschijnlijke uitspraak wijze, enzovoort) bestaan door de Radboud Universiteit geschreven scripts.

7.4.3 Stap 3: decompounden van de weinig voorkomende compounds

Weinig voorkomende compounds kunnen beter gesplitst worden om te zorgen dat de losse onderdelen 'robuuster`' worden. Het decompounden kan lastig zijn omdat a) sommige compounds op verschillende manieren gesplitst kunnen worden, b) sommige delen wel gesplitst kunnen maar niet moeten worden (bromden -> brom + den) en c) er problemen met de tussen **s** kunnen ontstaan (beoordelings**s**formulier -> beoordeling + s + formulier).

Ook voor het decompounden bestaan er door de Radboud Universiteit geschreven Perl-scripts.

7.4.4 Stap 4: genereren uitspraakwoordenboek.

Alle unieke woorden moeten fonetisch getranscribeerd worden middels een G2P. Er is een goede G2P beschikbaar op: https://webservices-lst.science.ru.nl/g2pservice

De woorden die niet in het achtergrondwoordenboek staan en waarvan de uitspraak dus gegenereerd wordt, moeten gecontroleerd worden.

7.4.5 Stap 5: herschrijven van de teksten

Wanneer alle teksten geschoond zijn, moeten de originele teksten herschreven worden voordat er een taalmodel gemaakt kan worden. Het decompounden en uniformeren van de schrijfwijze

gebeurt dikwijls op de woordenlijsten, maar de genomen stappen (mais -> maïs en 192 -> honderd twee en negentig) moeten dan in de teksten voor het taalmodel geëffectueerd worden. Dat kan met een simpele search & replace.

7.4.6 Stap 6: maken taalmodel

De meeste gebruikte manier om een taalmodel te maken is met de SRI-LM toolkit (zie: http://www.speech.sri.com/projects/srilm/)