Oplijning van tekst en geluid in Praat

Mart van Baalen
Studentnummer 6144136
Begeleider: David Weenink

Universiteit van Amsterdam
Faculteit Geesteswetenschappen
Afdeling Taalwetenschap

Abstract

In deze scriptie wordt de nauwkeurigheid en bruikbaarheid van automatische fonetische annotatie met behulp van een tekst-naar-spraaksysteem (En: Text-To-Speech, verder afgekort als TTS) besproken. Wat met annotatie wordt bedoeld wordt verder besproken in sectie 1.1 en paragraaf 1.3.1. Fonetische annotatie van spraaksignalen is in verschillende gebieden noodzakelijk. In taalwetenschappelijk onderzoek, bijvoorbeeld bij het beschrijven van het klinkersysteem van het Nederlands, spelen geannoteerde corpora een belangrijke rol, maar ook het trainen van spraakherkenningssystemen of het genereren van difonen voor difoonsynthese vereisen vele uren geannoteerde spraak (Holmes & Holmes, 2001). Het handmatig annoteren van een spraaksignaal is echter zeer tijdrovend. Als vuistregel wordt door fonetici aangehouden dat het annoteren van een signaal een factor 60 meer tijd kost dan de duur van het signaal: het annoteren van een seconde spraak neemt ongeveer een minuut tijd in beslag. Van Son et al. (2001) rapporteren een factor 30-50 voor het corrigeren van een grove voorannotatie van een spraaksignaal.

Om die reden zou het automatiseren van de annotatie voor veel partijen een uitkomst bieden. Praat (Boersma & Weenink, 2012) bevat sinds versie 5.3.05 een TTS-systeem. Met behulp van dit TTS-systeem en Dynamic Time Warping, een algoritme dat twee geluidssignalen met elkaar oplijnt, kan automatische annotatie plaatsvinden.

In deze scriptie worden de onderdelen van dit systeem besproken, en worden resultaten verzameld door middel van vergelijking van de automatische annotatie van dit systeem met handmatige annotatie van spraakcorpora. Deze resultaten worden vergeleken met andere systemen van automatische annotatie. Verder wordt een aantal suggesties ter verbetering van het systeem gedaan.

Inhoudsopgave

1 Inleiding **4**

1.1 Terminologie 4

1.2 Automatische annotatie 5

1.3 Literatuur 9

1.4 Vraagstelling 12

2 Onderdelen van het systeem **14**

2.1 Annotatie van spraaksignalen 14

2.2 Spraaksynthesizer 15

2.3 Minimum Edit Distance en String Alignment 16

2.4 Dynamic Time Warping 20

3 Werkwijze **31**

3.1 Annotatieconventies spraaksynthesizer 31

3.2 Gebruikte corpora 33

3.3 Vergaring Resultaten 35

3.4 Analyse resultaten 36

3.5 Evaluatie Resultaten 38

4 Resultaten **40**

4.1 Problemen met de output van het TTS-DTW algoritme in Praat 40

4.2 Totaalresultaten corpora 41

4.3 Verschillen tussen groepen in de corpora 43

4.4 Verschillen annotatie 45

4.5 Oorzaken van afwijkingen 51

5 Discussie en suggesties **58**

5.1 Discussie 58

5.2 Suggesties 59

5.3 Laatste opmerking 61

6 Conclusie **62**

Nawoord **63**

Literatuur **63**

Appendix A: Transcriptie-aanpassingen **67**

Appendix B: Editkostenmatrices **70**

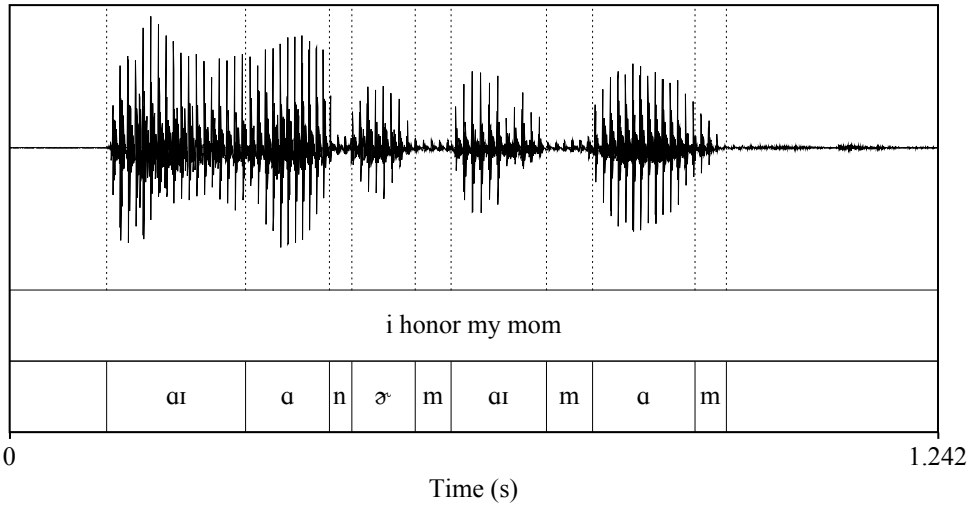
Appendix C: Scripts **71**

Hoofdstuk 1

Inleiding

1.1 Terminologie

In deze scriptie wordt een onderscheid gemaakt tussen annotatie en segmentering. Het segmenteren van een geluidssignaal betekent het opdelen in segmenten. In een spraaksignaal zijn dat doorgaans de fonemen, woorden of zinnen die in een signaal zijn uitgesproken. Praat (Boersma & Weenink, 2012) bevat een datatype, TextGrid, dat het segmenteren van een spraaksignaal mogelijk maakt. In figuur 1.1 is een in Praat gesegmenteerd en geannoteerd geluidssignaal te zien.



Figuur 1.1: Een opname van zin sx231, uitgesproken door spreker mtdb0 uit het TIMIT corpus (Lamel et al., 1986), geannoteerd in Praat met een TextGrid.

De annotatie van een spraaksignaal betekent het labelen van de segmenten van een spraaksignaal. Dit wil zeggen dat elk segment, in Praat genaamd 'interval', een label krijgt dat aanduidt wat de inhoud van het segment is.

Omdat annotatie een segmentering impliceert, wordt annotatie gebruikt als het onderscheid tussen de twee niet van belang is.

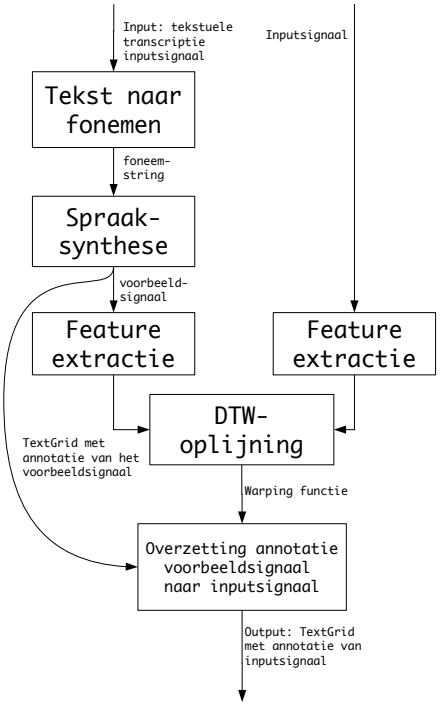
Daarnaast wordt een onderscheid gemaakt tussen overzetten en oplijnen. Met oplijnen wordt het vinden van overeenkomstige punten in reeksen bedoeld. Hoe dit in zijn werk gaat wordt beschreven in hoofdstuk 2. Met overzetten wordt bedoeld het omzetten van een annotatie van één signaal naar een annotatie van een ander signaal. Voor overzetten van een annotatie is oplijning van signalen nodig.

1.2 Automatische annotatie

In deze sectie wordt annotatie door een Text-To-Speech systeem en Dynamic Time Warping vergeleken met andere gangbare methodes van automatische annotatie. Daarna zal de keuze voor het hier besproken systeem worden verantwoord.

1.2.1 Automatische annotatie door een Text-To-Speech systeem en Dynamic Time Warping

Text-To-Speech (TTS) systemen synthetiseren een geluidssignaal uit tekst die door de gebruiker wordt ingevoerd. TTS systemen doen dit door een tekst om te zetten naar een fonetische representatie, en op basis van de fonetische representatie een geluidssignaal te genereren. Daarom moet een TTS systeem zelf aangeven waar de verschillende segmenten in een geluidssignaal beginnen en eindigen. Het gesynthetiseerde signaal (verder voorbeeldsignaal) wordt gebruikt om het te annoteren signaal (verder het inputsignaal) te annoteren, door de annotatie van het gesynthetiseerde signaal over te zetten naar een annotatie voor het inputsignaal. Figuur 1.2 geeft schematisch aan hoe het systeem werkt.



Figuur 1.2: Een schematische weergave van het TTS-DTW systeem, naar Malfrère & Dutoit (1997):2632.

Voor het kunnen voortbrengen van een perfecte annotatie moet een TTS-DTW systeem aan de volgende vier voorwaarden voldoen:

1. Om een spraaksignaal te kunnen synthetiseren uit een geschreven tekst, moet een TTS systeem de geschreven tekst omzetten naar een fonetische representatie, die bestaat uit reeks fonemen. Dit wordt verder de text-naar-fonemen stap genoemd. Om een

spraaksignaal correct[1] te kunnen annoteren, is het van belang dat de reeks fonemen die het TTS systeem genereert overeenkomt met de fonemen die in het inputsignaal aanwezig zijn.

Als de reeksen fonemen niet hetzelfde zijn, betekent dit dat de labels van sommige segmenten niet overeenkomen met de fonetische inhoud van dat segment. Als het TTS systeem een reeks fonemen voortbrengt met een ander aantal fonemen dan aanwezig zijn in het inputsignaal, heeft dit ook invloed op de segmentering van het inputsignaal.

Voorbeeld: In het TIMIT corpus (Lamel et al., 1986, zie sectie 3.2 voor meer informatie) wordt de shibbolethzin 'Don't ask me to carry an oily rag like that' door verchillende personen uitgesproken. Sommige van deze personen spreken de 'to' uit als [tu], anderen als [tə]. De spraaksynthesizer schrijft 'to' altijd om naar [tə]. Verder laten sommige sprekers de 't' in 'don't' weg. De spraaksynthesizer synthetiseert 'don't' altijd met een 't' aan het eind. In sommige gevallen verschilt de reeks fonemen die door het TTS systeem wordt voortgebracht van de reeks fonemen in het inputsignaal, waardoor het TTS-DTW systeem geen volledig correcte annotatie van het spraaksignaal zal voortbrengen.

2. De annotatie van de fonemen van het voorbeeldsignaal moet kloppen. Als de annotatie niet overeenkomt met de fonetische inhoud van het signaal, bijvoorbeeld als grenzen tussen foneemovergangen op een verkeerde plek worden geplaatst, wordt ook met een ideale overzetting van de annotatie van het voorbeeldsignaal naar een annotatie van het inputsignaal, het inputsignaal niet goed geannoteerd.

   Deze voorwaarde kan op een systematische manier worden overschreden. Hier wordt verder op ingegaan in secties 2.1 en 3.2.

3. De overzetting van de annotatie van het voorbeeldsignaal naar een annotatie van het inputsignaal moet correct plaatsvinden. Zoals later uitgebreid zal worden besproken, vindt de overzetting van de annotatie van het voorbeeldsignaal naar het inputsignaal plaats door middel van oplijning van het voorbeeldsignaal met het inputsignaal. Hieruit volgt dat nog een vierde voorwaarde geldt:

4. Het TTS systeem moet een voorbeeldsignaal synthetiseren dat dusdanig lijkt op het inputsignaal dat een acceptabele overzetting van de annotaties mogelijk is. Als het voorbeeldsignaal niet genoeg lijkt op het inputsignaal, zal het oplijnen van de twee signalen op problemen stuiten, waardoor de overzetting van de annotatie waarschijnlijk niet correct zal plaatsvinden.

[1] Over wat een correcte annotatie inhoudt kan uiteraard worden gediscussiëerd. In sectie 1.3.1 zal worden uitgelegd wat in deze scriptie wordt bedoeld met een correcte annotatie.

Deze voorwaarde staat niet volledig los van de eerste voorwaarde: als met een foneemstring die afwijkt van de fonemen aanwezig in het inputsignaal een voorbeeldsignaal wordt gesynthetiseerd, zal het voorbeeldsignaal zelf ook afwijken van het inputsignaal.

Voor de synthese van een voorbeeldsignaal wordt een spraaksynthesizer gebruikt. Deze is gebaseerd op eSpeak (Duddington, 2011), een TTS systeem dat op basis van fonetische en lexicografische regels een geluidssignaal synthetiseert. Doordat de spraaksynthesizer gebruik maakt van (een vorm van) formantsynthese, hoeft geen gebruik gemaakt te worden van vooraf opgenomen spraaksignalen, zoals wel nodig is voor difoon- of concatenatiesynthese. Punten 1, 2 en 4 van de hierboven besproken voorwaarden hebben betrekking op de het synthetiseren en annoteren van het voorbeeldsignaal, en dus op de spraaksynthesizer, en zullen verder worden besproken in hoofdstuk 2.

Voor de overzetting van de annotatie van het voorbeeldsignaal naar een annotatie van het inputsignaal wordt gebruik gemaakt van het Dynamic Time Warping (DTW) algoritme (Sakoe & Chiba, 1978). Een implementatie van DTW is in Praat aanwezig. Dit algoritme lijnt twee geluidssignalen met elkaar op door middel van lokaal rekken en krimpen van de geluidssignalen. Door te onthouden waar de geluidssignalen zijn gerekt en gekrompen, kunnen segmenten uit het voorbeeldsignaal naar het inputsignaal worden overgezet. Het DTW-algoritme wordt uitgebreider behandeld in hoofdstuk 2. De methode van automatische annotatie die in deze scriptie wordt gebruikt zal verder worden aangeduid met TTS-DTW.

1.2.2 Alternatieve manier van automatische annotatie: Spraakherkenning

De meest gebruikte methodes voor automatische annotatie zijn gebaseerd op spraakherkenningssystemen. De populairste methodes van spraakherkenning zijn gebaseerd op systemen die gebruik maken van Hidden Markov Models (HMM). Een Hidden Markov Model is een statistisch model waarmee de meest waarschijnlijke reeks fonemen voor een spraaksignaal kan worden bepaald.

Doordat op HMMs gebaseerde systemen segmentatie baseren op het spraaksignaal zelf en niet op een door Text-To-Speech regels gegenereerde foneemstring, kan een op HMMs gebaseerd segmentatiesysteem altijd een redelijk correcte segmentatie genereren, mits het trainingsmateriaal overeenstemt met het te annoteren signaal. De vier punten genoemd in paragraaf 1.2 gelden dus niet meer voor op HMMs gebaseerde systemen.

Aangezien HMMs statistische modellen zijn, moeten waarschijnlijkheden op de een of andere wijze worden bepaald. Bij HMMs gebeurt dit door te trainen op gesegmenteerde data. Voor elke taal waarin op HMMs gebaseerde systemen spraak moeten kunnen segmenteren is training op gesegmenteerde data nodig. Er bestaat een bootstrap methode,

genaamd forced alignment, waarmee ongesegmenteerde data waarvan de fonetische inhoud bekend is kan worden gebruikt om een HMM op te trainen (Jurafsky & Martin, 2009). Dit levert echter niet altijd een optimale annotatie op (Malfrère et al., 2003).

Op HMMs gebaseerde systemen kampen met het probleem dat overgangen tussen fonemen niet altijd op de juiste locatie worden geplaatst. Dit komt doordat het doel van op HMMs gebaseerde systemen is om de meest waarschijnlijke foneemstring te vinden, niet om de precieze lokatie van elk foneem te bepalen (Kominek et al., 2003).

HMMs zijn complexe modellen. Op dit moment zijn er geen op HMMs gebaseerde systemen voor automatische annotatie die gebruiksvriendelijk zijn voor onderzoekers die onbekend zijn met Hidden Markov Models.

1.2.3 Voordelen van TTS-DTW in Praat

In deze paragraaf wordt de keuze voor een TTS-DTW systeem verantwoord.

1. Het oplijnen van een voorbeeldsignaal met een inputsignaal gebeurt taalonafhankelijk. Op HMMs gebaseerde systemen moeten voor elke taal opnieuw worden getraind. Het DTW algoritme werkt voor elke taal hetzelfde. Hierdoor is het oplijnen makkelijk uit te breiden naar andere talen. Bovendien zit taalspecifieke kennis van het TTS-DTW systeem in het TTS systeem.

2. De spraaksynthesizer heeft geen difoon- of unitdatabase nodig om een spraaksignaal te synthetiseren. Doordat de spraaksynthesizer spraaksignalen synthetiseert op basis van regels, is ook voor het synthetiseren van het voorbeeldsignaal geen opgenomen spraak nodig. Voor het uitbreiden van de spraaksynthesizer naar een nieuwe taal is alleen kennis van het fonetisch systeem van een taal nodig.

   Dit punt is relevant, omdat alle andere op dit moment beschikbare systemen die TTS-DTW toepassen gebruik maken van difoonsynthese of unit selectie synthese. Hierdoor is veel segmenteerde data nodig om deze systemen uit te breiden naar nieuwe talen. Dit is niet het geval voor de implementatie van TTS-DTW in Praat. In paragraaf 1.3.4 worden andere systemen die TTS-DTW implementeren uitgebreider besproken.

3. De op eSpeak gebaseerde spraaksynthesizer in Praat bevat 41 ingebouwde talen. Voor een aantal talen kan ook een dialect gekozen worden. Sommige van deze talen zijn nog niet uitvoerig getest, maar bieden in ieder geval een aanknopingspunt voor verdere ontwikkeling. eSpeak is om deze reden een goede keuze om een rule-based spraaksynthesizer in Praat op te baseren.

4. De TTS-DTW methode zoals deze in Praat wordt geïmplementeerd, zal gebruiksvriendelijk zijn. Nu beschikbare implementaties van deze methode zijn gericht op onderzoek

naar spraaksynthese. Hierdoor ontbreekt het in deze methodes aan gebruiksvriendelijkheid en bruikbaarheid voor fonetisch onderzoek. Door de incorporatie in Praat kan het gesegmenteerde signaal gelijk voor verdere analyse worden gebruikt.

5. Praat is volgens Jurafsky & Martin (2009) het populairste programma voor fonetische analyse. Dit betekent dat incorporatie van het TTS-DTW systeem in Praat het systeem bereikbaar maakt voor een groot aantal onderzoekers.

1.3 Literatuur

In deze sectie worden de resultaten van andere systemen voor automatische annotatie besproken. Met name worden andere TTS-DTW sytemen besproken. Op basis hiervan zal in de volgende sectie een hypothese over de werking van het TTS-DTW in Praat worden opgesteld. Verder zullen recente systemen van automatische annotatie worden besproken, zodat bekend is hoe goed deze op dit moment werken. Verder wordt een aantal programma's die een implementatie van het TTS-DTW algoritme bevatten besproken.

1.3.1 Een acceptabele annotatie

In deze scriptie zullen de annotaties van het TTS-DTW systeem worden vergeleken met handmatige annotaties. Dit gebeurt door de afwijking tussen foneemgrenzen tussen de automatische annotatie en de handmatige annotatie te bepalen. De precieze werkwijze hiervan staat in hoofdstuk 3. Om te kunnen stellen dat een annotatiesysteem een goede annotatie levert, moet eerst worden geëxpliciteerd wat een goede annotatie is. Voor zover dit gebeurt in de onderstaande onderzoeken, wordt verwezen naar onderzoek van Cosi et al. (1991). In dit onderzoek worden geluidssignalen geannoteerd door fonetische experts. Elk geluidssignaal wordt door meerdere experts geannoteerd. De annotaties worden met elkaar vergeleken door de afwijking van grenzen tussen foneemovergangen tussen experts te bepalen.

Cosi et al. (1991) vinden een gemiddelde afwijking van 7 ms. Daarnaast waren tussen de 61% en 100% van de afwijkingen onder de 20 milliseconden (ms).

Vanwege de niet-symmetrische verdeling van de afwijkingen en de grote outliers die kunnen voorkomen, is de gemiddelde afwijking geen goede maat om te bepalen hoe goed het TTS-DTW systeem werkt. Berekenen hoeveel grenzen tussen fonemen een afwijking hebben die kleiner is dan een bepaalde grens is een zinvollere maat, omdat duidelijk wordt hoeveel grenzen handmatig zouden moeten worden gecorrigeerd om een annotatie te krijgen die binnen de te verwachten afwijking van een handmatige annotatie blijft.

Een acceptabele annotatie is een annotatie die een handmatige annotatie benadert. Aangezien de afwijking tussen experts voor 61%-100% van de grenzen onder de 20 ms blijft, zal wordt dit aangehouden als een acceptabele annotatie.

1.3.2 TTS-DTW annotatie

De methode van automatische annotatie die in deze scriptie wordt geëvalueerd is eerder gebruikt. Malfrère & Dutoit (1997) onderzochten als een van de eersten de mogelijkheden van automatische annotatie middels een TTS-DTW systeem. Zij gebruikten hiervoor MBROLA (Dutoit et al., 1996), een TTS systeem dat gebruik maakt van difoonsynthese voor het genereren van een geluidssignaal. De auteurs rapporteren redelijk goede resultaten (56.3% - 96.1% van de afwijkingen van de annotatie van grenzen tussen fonemen <20ms, gemiddeld 78.2%, resultaten afgesplitst naar de spreker van wie het inputsignaal afkomstig was), maar vermelden niet wat de afwijkingen veroorzaakt.

Horák (2001) evalueert de bruikbaarheid van het TTS-DTW annotatiesysteem voor het Tsjechisch. Voor het TTS systeem werd Epos gebruikt, een difoonsynthesizer. Zijn doel is het verkrijgen van een difoondatabase voor gebruik in een difoonsynthesizer. Horák stelt dat de op deze wijze verkregen annotatie nauwkeurig genoeg is voor het extraheren van intonatiecontouren, maar niet nauwkeurig genoeg voor het genereren van een difoondatabase. Hiervoor zou na de TTS-DTW annotatie met handmatige nacontrole de annotatie worden verbeterd. Horák rapporteert redelijk goede resultaten (64.5% - 63.3% van de afwijkingen in de annotatie van de onsets van fonemen <20 ms, resultaten afgesplitst naar foneem), maar zijn resultaten zijn slechter dan die van Malfrère & Dutoit (1997).

In Malfrère et al. (2003) wordt de TTS-DTW methode verder onderzocht en vergeleken met annotatie op basis van een systeem gebaseerd op een HMM. De vergelijkingen zijn gebaseerd op afwijkingen ten opzichte van handmatige annotatie van verschillende corpora. De tests zijn uitgevoerd op een Amerikaans-Engels corpus (TIMIT), een Nederlands corpus (COGEN), een Frans corpus (BDSONS) en een Spaans corpus (LATINO-40). De verschillen in afwijking tussen de HMM en de TTS-DTW methodes zijn verwaarloosbaar klein voor Frans en Engels (58.6% - 81.7% van de afwijkingen in de annotatie van de overgangen tussen fonemen <20 ms. Resultaten opgesplitst naar klinker en consonanten). TTS-DTW geeft echter aanzienlijk slechtere resultaten op de Nederlandse en Spaanse corpora. De auteurs wijden dit aan een groter aantal klinker-klinker overgangen in deze talen.

In dit artikel stellen de auteurs een hybride systeem voor waarin DTW-oplijning wordt gebruikt om een HMM te trainen.

Voor TTS-DTW systemen gebaseerd op difoonspraaksynthesizers melden Malfrère & Dutoit (1997) en Malfrère et al. (2003) dat betere prestaties worden behaald als voor annotatie van spraaksignalen van mannen een mannelijke synthesestem wordt gebruikt, en voor annotatie van spraaksignalen van vrouwen een vrouwenstem.

In Kominek et al. (2003) wordt de TTS-DTW methode zoals deze is geïmplementeerd in Festvox (Black & Lenzo, 2007) besproken. De TTS-DTW methode in Festvox is gebaseerd op de methode beschreven in Malfrère & Dutoit (1997), met synthese gebaseerd op de

concatenatie van segmenten van opgenomen spraak. In Kominek et al. (2003) wordt de TTS-DTW methode vergeleken met een op Hidden Markov Models gebaseerde methode. Hieruit blijkt dat de TTS-DTW methode in Festvox erg gevoelig is voor niet-matchende pauzes, pauzes die alleen in het voorbeeldsignaal of alleen in het inputsignaal, maar niet in beide signalen voorkomen. De auteurs merken op dat Festvox (Black & Lenzo, 2007, zie ook de volgende sectie), in de TTS stap, soms andere fonemenen realiseert dan de fonemen aanwezig in het spraaksignaal. Het TTS-DTW systeem vertoont echter in 70% van de grenzen tussen fonemen een kleinere afwijking ten opzichte van handmatige annotatie dan het HMM systeem. De afwijkingen van slechtste annotatie van het TTS-DTW systeem zijn echter veel groter dan de afwijkingen van de slechtste annotaties van de op HMMs gebaseerde systemen.

Latere onderzoeken laten slechtere prestaties van TTS-DTW systemen zien ten opzichte van HMM systemen of hybride systemen. Adell et al. (2005) zien in hun artikel waarin verschillende methodes voor automatische annotatie worden besproken dusdanig slechte resultaten met een TTS-DTW systeem dat zij deze methode niet opnemen in hun vergelijking van verschillende automatische annotatiemethoden.

1.3.3 Andere methodes

Jakovljević et al. (2012) beschrijven een methode van automatische annotatie gebaseerd op HMMs. Voor hun systeem is geen vooraf geannoteerd corpus nodig, maar wel een transcriptie van de tekst. Het systeem maakt gebruik van statistische nacorrectie, waarbij grenzen op de meest waarschijnlijke plek worden gezet. Hiervoor wordt spectrale informatie gebruikt. Jakovljević et al. (2012) rapporteren een afwijking ten opzichte van een handmatig geannoteerd Hebreeuws corpus van <5ms voor 52.5% van de grenzen tussen fonemen, <10ms voor 75.8% van de grenzen tussen fonemen en 90.6% van <20ms van de grenzen tussen fonemen.

Keshet et al. (2007) maken gebruik van een systeem gebaseerd op Support Vector Machines, een algoritme dat data in twee groepen scheidt. Hiermee bereiken zij op het TIMIT corpus een afwijking ten opzichte van de handmatige annotatie van <10ms voor 79.9% van de grenzen tussen fonemen en <10ms voor 92.3% van de grenzen tussen fonemen.

1.3.4 Beschrijving van programma's die gebruik maken van een TTS-DTW annotatiesysteem

Zoals al vermeld in paragraaf 1.2.3 zijn alle nu beschikbare TTS-DTW systemen gericht op onderzoek naar spraaksynthese. De onderstaande TTS-DTW systemen zijn alle ontwikkeld voor het genereren van synthesestemmen of voor het extraheren van intonatiecontouren voor difoonsynthese of unit-selection synthese.

MBROLIGN

MBROLIGN (Dutoit, 1999) is ontwikkeld door Thierry Dutoit. Het programma is een uitvloeisel van het onderzoek beschreven in Malfrère & Dutoit (1997). Het programma maakt gebruik van MBROLA (Dutoit et al., 1996) stemmen voor de synthese van het voorbeeldsignaal. Dit programma was oorspronkelijk ontwikkeld om het genereren van nieuwe stemmen voor difoonsynthese of unit selectie synthese te vergemakkelijken. Het systeem wordt nu echter alleen nog gebruikt voor het extraheren van intonatiecountouren.

Festival/FestVox

Festival (Taylor et al., 1998) is een modulair TTS systeem, oorspronkelijk ontwikkeld door Alan W. Black, Paul Taylor en Richard Caley aan de Universiteit van Edinburgh in het Centre for Speech Technology Research. Festival wordt nu onderhouden door Black aan de Carnegie Mellon University.

Festvox, een onderdeel van Festival, ondersteunt het bouwen van stemmen voor difoonsynthese en unit selectie synthese (Black & Lenzo, 2007). Verder bevat Festvox een op Hidden Markov Models gebaseerde manier van annotatie, SphinxTrain (Seltzer & Singh, 2012).

Eposligner

Eposligner is een systeem vergelijkbaar met MBROLIGN (Dutoit, 1999) en wordt beschreven in Horák (2001). Ten tijde van het schrijven van deze scriptie was de website van Eposligner niet meer online, en kon het programma niet meer worden gedownload.

Hier moet worden opgemerkt dat de bovenstaande TTS-DTW systemen alle niet meer actief onderhouden worden. MBROLIGN is niet meer geüpdate sinds 2002, Festival/Festvox maakt voornamelijk gebruik van op HMMs gebaseerde systemen en over Eposligner is, buiten Horák (2001), vrijwel geen informatie te vinden.

1.4 Vraagstelling

In de eerste plaats zal in deze scriptie de bruikbaarheid van het hierboven besproken automisch annotatiesysteem worden onderzocht. Hiermee wordt bedoeld dat bekeken wordt of gebruik van een TTS-DTW systeem tijdswinst voor het annotatieproces kan opleveren. Op basis van de hierboven geciteerde literatuur is dit zeer waarschijnlijk.

Daarnaast zal worden onderzocht of TTS-DTW handmatige annotatie volledig overbodig kan maken. Op basis van de literatuur hierboven is het waarschijnlijk dat volledig acceptabele annotatie door een TTS-DTW systeem altijd gepaard zal moeten gaan met een handmatige nacontrole.

Ten slotte zal worden onderzocht waardoor problemen in de automatische annotatie worden veroorzaakt, en hoe het systeem in de toekomst kan worden verbeterd. De literatuur laat vooral bij klinker-klinker overgangen en niet-matchende stiltes problemen zien bij het toepassen van de TTS-DTW systeem. Verwacht wordt dat niet voldoen aan de voorwaarden besproken in paragraaf 1.2.1 problemen zal opleveren in de automatische annotatie.

Hoofdstuk 2

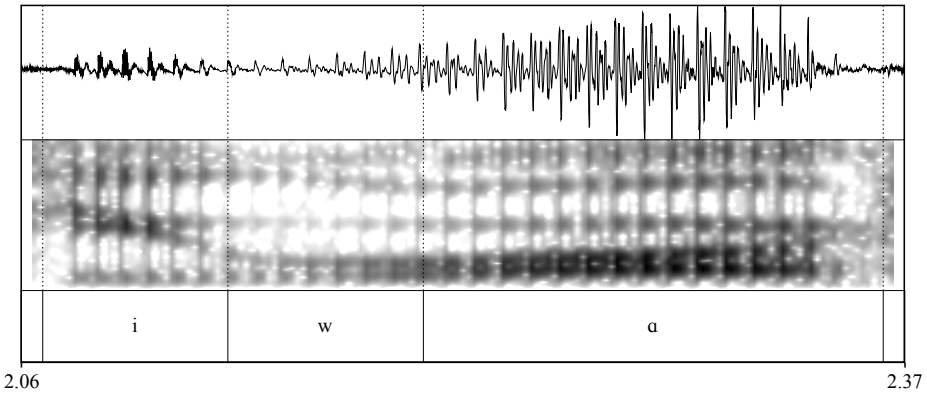
Onderdelen van het systeem

In dit hoofdstuk wordt verder ingegaan op de details van de spraaksynthesizer, het string alignment algoritme en het dynamic time warping algoritme.

2.1 Annotatie van spraaksignalen

Fonen in continue spraak zijn niet van foon tot foon stabiel. Fonen worden als het ware besmet door één of meerdere voorgaande en volgende fonen. Als een Engelsspreker /iwɑ/ zegt, zal een luisteraar drie verschillende fonemen horen, de /i/, de /w/ en de /ɑ/. Deze vormen in het spraaksignaal echter geen discrete eenheden: zie figuur 2.1. In deze figuur, afkomstig uit TIMIT en bestaande uit een woord uit zin sa1, uitgesproken door spreker mbwm0, zijn de drie fonemen in elk in een constante staat van transitie. Dit betekent dat in het annoteren van een spraaksignaal geen absolute grens is te trekken tussen twee fonemen, en dat keuzes moeten worden gemaakt over wat een wenselijke annotatie is. Om deze reden worden daarom bij de constructie van grote spraakcorpora conventies afgesproken voor het consistent annoteren van spraaksignalen.

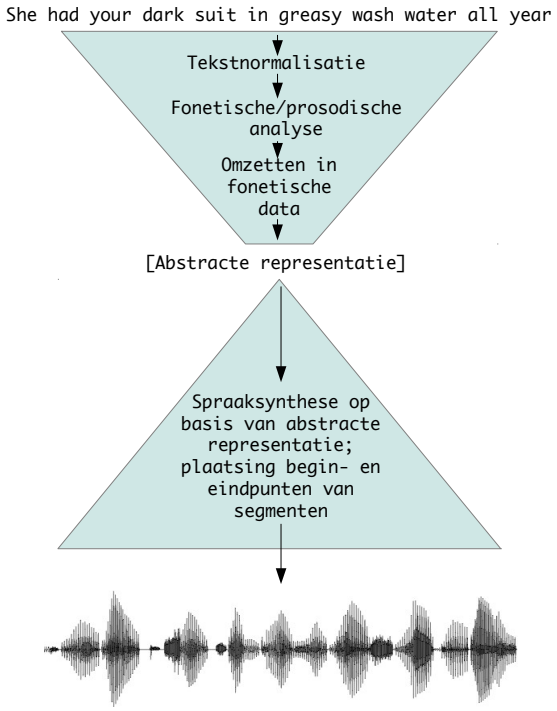
Daarnaast moet worden gekozen welke karakterset wordt gebruikt. Het internationaal fonetisch alfabet (IPA) biedt de meeste flexibiliteit en nauwkeurigheid, maar wordt niet door elk computersysteem ondersteund. Daarom zijn in de loop der tijd verschillende transcripties van IPA symbolen ontwikkeld op basis van ASCII symbolen, die op vrijwel alle computersysteem kunnen worden gebruikt. Voorbeelden van complete transcriptie van het IPA in ASCII symbolen zijn de Kirshenbaumtranscriptie (Kirshenbaum, 2001) en de SAMPA transcriptie (Wells, 1997). Ook bestaan systemen die een transcriptie voor één enkele taal mogelijk maken, zoals het Arpabet. Deze bevatten geen transcriptie voor alle IPA symbolen, maar alleen voor de spraakklanken die in de getranscribeerde taal voorkomen. In het geval van Arpabet is dat het Engels.



Figuur 2.1: De fonemen /iwɑ/ afkomstig uit zin sa1 uit het TIMIT corpus (Lamel et al., 1986), uitgesproken door de mannelijke spreker mbwm0. Zin sa1 is de shibbolethzin 'She had your dark suit in greasy wash water all year'. De fonemen in de annotatie komen overeen met de 'y' in greasy en de 'w' en 'a' uit 'wash'.

2.2 Spraaksynthesizer

Praat bevat sinds versie 5.3.05 een spraaksynthesizer, welke is gebaseerd op eSpeak (Duddington, 2011). Het genereren van gesynthetiseerde spraak vindt in de spraaksynthesizer, conform de zandlopermetafoor van (Taylor, 2009, in Jurafsky & Martin (2009)), in twee stappen plaats (zie ook figuur 2.2). In de eerste stap wordt, op basis van orthografische, fonetische en prosodische regels voor een taal, een tekst in die taal omgezet in een foneemstring. De orthografische regels bepalen hoe woorden door fonemen moeten worden weergegeven. Een orthografische regel voor het Nederlands zou er zo uit zien: oe → [u]. Ook langere sequenties van letters, zoals het suffix -lijk, worden opgenomen in de orthografische regels. Ongebruikelijk gespelde woorden staan in een uitspraakwoordenboek. Voor het Nederlands zou een woord als Gorinchem, waarvan de uitspraak /χɔrkəm/ door



Figuur 2.2: De zandlopermetafoor van Taylor (2009), naar Jurafsky & Martin (2009):284.

middel van orthografische regels niet kan worden afgeleid uit de geschreven vorm, in een woordenboek moeten worden opgenomen.

De fonetische en prosodische regels bepalen hoe de foneemstring die door de orthografische regels is gegenereerd in een geluidssignaal moet worden omgezet. Fonetische regels bevatten informatie over de formanten van sonoranten. In het voorbeeld van de [u] zou een fonetische regel er zo uitkunnen zien: [u] $\rightarrow F_1 = 320Hz; F_2 = 800Hz$. Verder worden in de fonetische regels gepaste lengtes van fonemen en coarticulatieregels voor fonemen binnen een woord gespecificeerd. In de prosodische regels worden F_0-bewegingen gespecificeerd.

Beide sets van regels zijn toegankelijk en kunnen bewerkt worden. Bovendien kan de gebruiker zelf een taal toevoegen aan het repertoire van de spraaksynthesizer door fonetische, prosodische en orthografische regels te specificeren.

In de tweede stap wordt de uit de eerste stap verkregen data vertaald naar gesynthetiseerde spraakgeluiden. De spraaksynthesizer gebruikt formantsynthese om spraak te synthetiseren. Dit betekent dat er geen gebruik wordt gemaakt van vooraf opgenomen geluidsfragmenten. Een voordeel hiervan is dat alleen de hierboven besproken regels nodig zijn voor spraaksynthese, waardoor de omvang van de voor de spraaksynthesizer benodigde data beperkt blijft. Verder vergemakkelijkt dit het toevoegen van een taal, omdat hiervoor alleen kennis van het fonetisch systeem van een taal beschikbaar moet zijn, en geen opgenomen spraak nodig is.

Voor de fonemische transcriptie van het gesynthetiseerde signaal gebruikt de spraaksynthesizer de ASCII compatibele Kirshenbaumtranscriptie (Kirshenbaum, 2001).

2.3 Minimum Edit Distance en String Alignment

In deze scriptie zal de werking van het TTS-DTW systeem in Praat worden getest aan de hand van vergelijkingen met handmatige annotaties van spraaksignalen. Eerder is gesteld dat het belangrijk is dat in de tekst-naar-fonemen stap de fonemen worden gerealiseerd die ook aanwezig zijn in het inputsignaal. Als dit niet het geval is, wat door de hoge variabiliteit van het spraaksignaal mogelijk is, komen in de automatische annotatie andere segmenten voor dan in de handmatige. Zo kan bijvoorbeeld een segment in de automatische annotatie zijn ingevoegd of verwijderd ten opzichte van de handmatige annotatie. Hierdoor is het waarschijnlijk dat fonemen in de TTS-DTW annotatie van het inputsignaal niet op dezelfde plek staan als in de handmatige annotatie van het inputsignaal. Om bij elkaar behorende fonemen te vinden, wordt gebruik gemaakt van het String Alignment algoritme (Jurafsky & Martin, 2009).

Het string alignment algoritme (Jurafsky & Martin, 2009) is een algoritme waarmee twee reeksen tekens (Eng: string) met elkaar kunnen worden opgelijnd. Een maat om de afstand tussen twee strings te kwantificeren is de minimale bewerkingsafstand (Eng: Minimum Edit

intention	
	← *deletie i*
ntention	
	← *substitutie n door e*
etention	
	← *substitutie t door x*
exention	
	← *insertie c*
execntion	
	← *substitutie n door u*
execution	

Tabel 2.1: Operaties uitgevoerd op bronwoord 'INTENTION' om tot doelwoord 'EXECUTION' te komen, naar Jurafsky & Martin (2009):109

Distance, verder MED). De MED tussen twee strings wordt gegeven door het kleinste aantal operaties dat nodig is om van een bedoelde string tot een geobserveerde string te komen te tellen. De bedoelde string wordt bron of bronstring genoemd en de geobserveerde string doel of doelstring Jurafsky & Martin (2009). Geldige operaties zijn het invoegen van een teken in het bronwoord (insertie), het verwijderen van een teken uit het bronwoord (deletie) en het vervangen van een teken in het bronwoord (substitutie). In tabel 2.1 is te zien welke operaties nodig zijn op bronwoord 'intention' om tot doelwoord 'execution' te komen.

Door gewichten toe te kennen aan elke operatie kan invloed op de bewerkingsafstand worden uitgeoefend. De MED is dan de keuze van operaties die de kleinste cumulatieve afstand heeft. De cumulatieve afstand wordt berekend door de weegfactoren van alle uitgevoerde operaties bij elkaar op te tellen. Vaak worden insertie en deletie gewogen met een factor 1, en substitutie met een factor 2, aangezien substitutie kan worden worden bereikt door het achtereenvolgens uitvoeren van een deletie en een insertie. Met deze weegfactoren is de cumulatieve afstand van de operaties in tabel 2.1 in totaal 8.

In de oplijning van handmatige annotaties met automatische annotaties is de handmatige annotatie de annotatie die gewenst is, en de automatische annotatie de waargenomen annotatie. De handmatige annotatie wordt dus gezien als bronannotatie en de automatische annotatie als doelannotatie.[1]

Deze weegkosten kunnen verder worden verfijnd door de introductie van een editkostenmatrix, waarmee de weegkosten per bron- en doelteken geregeld kunnen worden. In figuur 2.3 is een kleine editkostenmatrix te zien.

Het vinden van de reeks operaties die de kleinste cumulatieve afstaand heeft is geen triviaal probleem. De reeks operaties die is beschreven in tabel 2.1 is niet de enige reeks operaties die 'intention' kan transformeren in 'execution'. Door de de letters 'inten' te verwijderen, de letters 'tion' te substitueren door 'exec' en de letters 'ution' in te voegen

[1] In Praat is deze conventie precies andersom: de bronstring is de observatie en de doelstring is de bedoelde string.

	a	b	c	d	e	[del]
a	0	2	2	2	1	0.5
b	2	0	2	1.5	2	1
c	2	2	0	2	2	1
d	2	1.5	2	0	2	1
e	1	2	2	2	0	1
[ins]	1	1	1.5	1	1	0

Figuur 2.3: Een editkostenmatrix voor een alfabet van 5 tekens. Het bronalfabet staat in deze tabel aan het begin van elke rij, en het doelalfabet bovenaan elke kolom. Het bronalfabet is in dit voorbeeld hetzelfde als het doelalfabet. De rij met [ins] en de kolom met [del] geven de weegfactor van respectievelijk insertie van tekens uit het doelalfabet en deletie van tekens uit het bronalfabet weer.

Tabel 2.2: Een bewerkingsafstandsmatrix. De getallen in de cellen geven de kortste afstand tot het bronwoord weer. De pijlen in de cellen geven aan vanuit welke vorige cel het pad naar de huidige cel kan zijn gelopen. De lichtgrijs gekleurde cellen geven het uiteindelijk gekozen kortste pad van het bronwoord naar het doelwoord. Uit: Jurafsky & Martin (2009):111

n	9	↓ 8	↙↓← 9	↙↓← 10	↙↓← 11	↙↓← 12	↓ 11	↓ 10	↓ 9	↙ 8
o	8	↓ 7	↙↓← 8	↙↓← 9	↙↓← 10	↙↓← 11	↓ 10	↓ 9	↙ 8	← 9
i	7	↓ 6	↙↓← 7	↙↓← 8	↙↓← 9	↙↓← 10	↓ 9	↙ 8	← 9	← 10
t	6	↓ 5	↙↓← 6	↙↓← 7	↙↓← 8	↙↓← 9	↙ 8	← 9	← 10	← 11
n	5	↓ 4	↙↓← 5	↙↓← 6	↙↓← 7	↙↓← 8	↙↓← 9	↙↓← 10	↙↓← 11	↙↓ 10
e	4	↙ 3	← 4	↙← 5	← 6	← 7	←↓ 8	↙↓← 9	↙↓← 10	↓ 9
t	3	↙↓← 4	↙↓← 5	↙↓← 6	↙↓← 7	↙↓← 8	↙ 7	←↓ 8	↙↓← 9	↓ 8
n	2	↙↓← 3	↙↓← 4	↙↓← 5	↙↓← 6	↙↓← 7	↙↓← 8	↓ 7	↙↓← 8	↙ 7
i	1	↙↓← 2	↙↓← 3	↙↓← 4	↙↓← 5	↙↓← 6	↙↓← 7	↙ 6	← 7	← 8
#	0	1	2	3	4	5	6	7	8	9
	#	**e**	**x**	**e**	**c**	**u**	**t**	**i**	**o**	**n**

wordt het doelwoord ook bereikt. Op deze manier is een zeer groot aantal andere mogelijke reeksen operaties te vinden die ‘intention’ omschrijven naar ‘execution’. Al deze manieren zijn echter kostbaarder dan de reeks operaties beschreven in tabel 2.1.

De operaties die worden uitgevoerd op een bronwoord kunnen worden gezien als een pad van het bronwoord naar het doelwoord, waarin elke stap op het pad overeenkomt met een bepaalde operatie (insertie, substitutie of deletie). Elk woord in tabel 2.1 komt dus overeen met een stap op het pad van ‘intention’ naar ‘execution’. Het aantal mogelijke paden tussen de beide woorden is zeer groot. Het kortste pad kan niet efficiënt worden gevonden door alle mogelijke paden uit te proberen en het kortste pad te kiezen. Het string alignment algoritme maakt gebruik van een techniek genaamd Dynamisch Programmeren om de reeks operaties met de laagste kosten te vinden.

Dynamisch programmeren is een klasse algoritmes die een moeilijk oplosbaar hoofdprobleem opdelen in makkelijker oplosbare subproblemen. Het hoofdprobleem kan worden opgelost door oplossingen voor de subproblemen te combineren. Het hoofdprobleem hier is het vinden van het pad met de laagste kosten tussen bronwoord en doelwoord. Met dynamisch programmeren wordt dit als volgt opgelost.

Stel dat een woord op het optimale pad tussen bronwoord en doelwoord ligt, bijvoorbeeld ‘exention’. Het pad van het bronwoord naar ‘exention’ moet onderdeel zijn van het totale kortste pad. Als dit niet het geval was, zou er een korter pad naar dit woord bestaan, waardoor er een korter pad van bronwoord naar doelwoord gevonden kon worden. Dit geldt ook voor alle woorden tussen het bronwoord en ‘exention’ (Jurafsky & Martin, 2009).

Het kortste pad tussen het bronwoord en het doelwoord wordt berekend door een bewerkingsafstandsmatrix op te stellen. Voor de bewerkingsafstandsmatrix geldt dat langs de rijen tekens uit het bronwoord staan, en langs de kolommen tekens uit het doelwoord (zie tabel 2.2). Elke cel in de bewerkingsafstandsmatrix geeft aan hoeveel het minimale pad vanaf het bronwoord tot aan die cel kost. Inserties vanuit het bronwoord komen overeen met horizontale stappen in de matrix, deleties komen overeen met verticale stappen en substituties komen overeen met schuine stappen, waarbij het teken van het bronwoord op die plek niet overeenkomt met het teken van het target.

Het kortste pad tussen het bronwoord en een woord A dat op het kortste pad tussen een woord B ligt, is onderdeel van het kortste pad tussen het bronwoord en B. Daarom kan het kortste pad van het bronwoord tot een bepaalde cel worden bepaald door te kijken vanuit welke vorige cel die cel met de laagste kosten kan worden bereikt. Om dit efficiënt te kunnen berekenen, gelden voor het pad tussen het bronwoord en het doelwoord twee voorwaarden:

1. Het pad moet monotoon zijn: als de matrix wordt gevisualiseerd als in tabel 2.2 loopt het pad van linksonder naar rechtsboven. Elke stap moet het pad dichter bij het doelwoord brengen: het pad moet dus altijd naar rechts, naar boven, of naar rechts en naar boven lopen.

2. Het pad moet continu zijn: het pad naar een element moet afkomstig zijn van een element direct links van, direct onder of direct linksonder het element.

3. Het pad moet vanaf het begin van beide woorden naar het eind van beide woorden lopen. In de matrix betekent dit dat het linksonder moet beginnen, en niet mag eindigen voordat het element rechtsboven is bereikt.

Het algoritme werkt nu als volgt. Als I het aantal tekens in de source is, en J het aantal tekens in de target, wordt een $(I + 1) \times (J + 1)$ matrix voor de bewerkingsafstand opgesteld. In het voorbeeld in tabel 2.2 is de matrix vanaf 0 geïndexeerd, met het element 0,0 linksonder. Het element 0,0 krijgt waarde 0, aan het begin zijn er immers nog geen operaties op het source woord uitgevoerd.

Voor elke cel i, j worden de kosten van het kortste pad van de source tot aan cel i, j, waarbij i de index van een teken uit de source is, en j de index van een teken uit de target,

nu als volgt berekend[2]:

$$distance_{i,j} = \min \left(\begin{array}{l} distance_{i-1,j} + \text{ins-cost}(target_j) \\ distance_{i-1,j-1} + \text{sub-cost}(source_i, target_j), \\ distance_{i,j-1} + \text{del-cost}(source_i), \end{array} \right) \tag{2.1}$$

Dit betekent dat de minimale cumulatieve kosten om vanuit bron tot aan cel i, j te komen gelijk zijn aan het minimum van de som van de minimale cumulatieve kosten van elk van de drie mogelijk cellen van waaruit de huidige cel kan worden bereikt en de kosten van de operatie om van de vorige cel in de huidige cel te komen.

Oplijning van strings vindt plaats door in elke huidige cel te onthouden vanuit welke vorige cel de huidige cel werd bereikt met de laagste cumulatieve kosten. Als de huidige cel vanuit meerdere cellen met even lage cumulatieve kosten kan worden bereikt, wordt een voorkeur voor schuine stappen aangehouden. In elke cel wordt een backpointer bijgehouden, die aangeeft vanuit welke cel de huidige cel werd bereikt. Door vanuit de cel rechtsboven in de matrix de backpointers terug te volgen totdat de cel linksonder is bereikt, is het pad door de matrix met de kortste cumulatieve afstand gevonden.

Uit dit pad kan worden afgeleid waar inserties, deleties en substituties hebben plaatsgevonden. Zo kan uit het pad in tabel 2.2 bijvoorbeeld worden afgeleid dat de 'i' uit 'intention' wordt verwijderd, en dat de 'n' wordt opgelijnd met de eerste 'e' van 'exectuion'.

2.4 Dynamic Time Warping

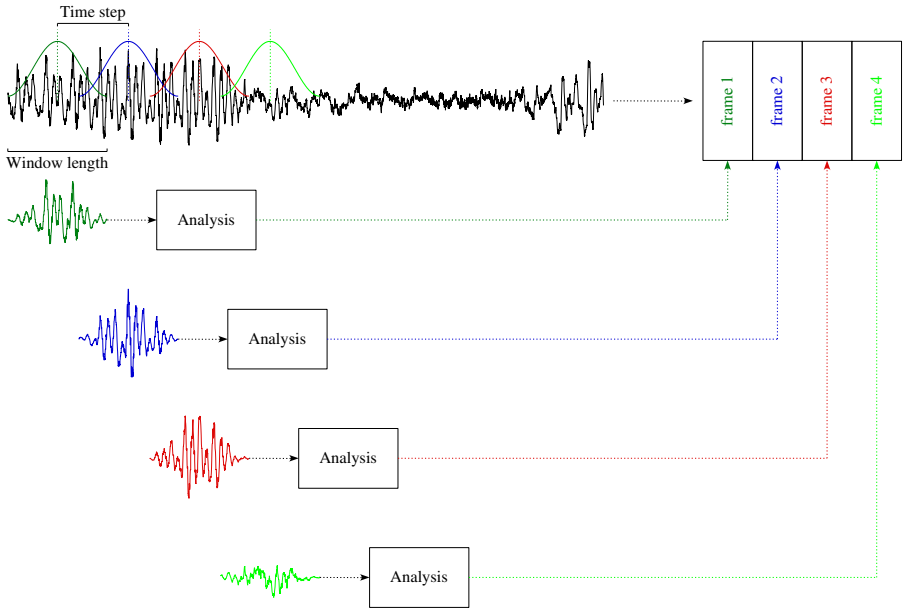
Zoals vermeld in de inleiding, wordt het DTW algoritme gebruikt om een annotatie van het voorbeeldsignaal te vertalen naar het inputsignaal. Dit doet het DTW algoritme door het inputsignaal op te lijnen met het voorbeeldsignaal. Het inputsignaal dient hier als bronsignaal.

De oplijning vindt plaats op een manier die sterk lijkt op het string alignment algoritme. In plaats van reeksen tekens worden in het DTW algoritme reeksen analyseframes met elkaar opgelijnd. Analyseframs zijn korte, geanalyseerde fragmenten van het bron- en doelsignaal. Het verkrijgen van analyseframes uit de beide signalen gebeurt in een voorbewerkingsfase met behulp van het Mel Frequency Cepstral Coefficients algoritme. Met de analyseframes wordt, net als in het string alignment algoritme, een bewerkingsafstandsmatrix opgebouwd, waarin het pad met de kortste cumulatieve afstand wordt bepaald.

Hieronder wordt elke stap in de voorbewerking besproken. Daarna wordt de oplijning behandeld.

[2]In Jurafsky & Martin (2009) wordt de conventie dat de eerste index de rijen aangeeft en de tweede index de kolommen gebroken

2.4.1 Voorbewerking

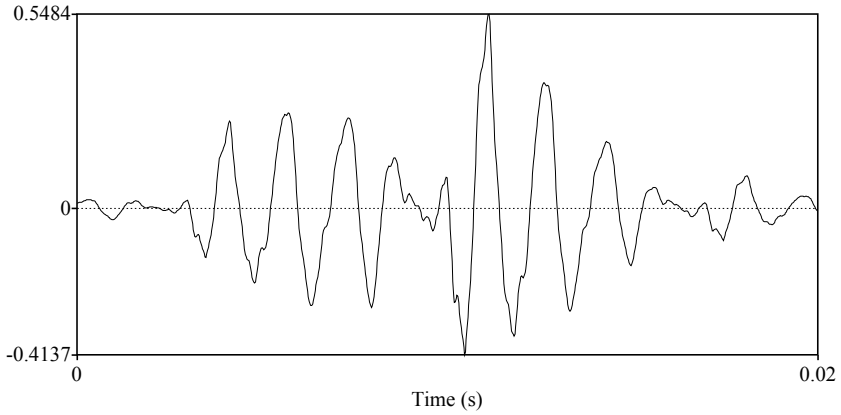


Figuur 2.4: De voorbewerking vn een geluidsignaal. De gekleurde lijnen geven de vensterfunctie aan voor elk analysevenster. De analyse die op elk analysevenster wordt uitgevoerd gebeurt met MFCCs. Uit: Weenink (unpublished):22.

In figuur 2.4 staat een schematisch overzicht van voorbewerking van geluidssignalen afgebeeld. Deze methode van voorbewerking wordt voor DTW, maar ook voor vele andere analysemethodes gebruikt. In deze figuur is te zien dat een geluidssignaal wordt gevensterd. Elk analysevenster wordt afzonderlijk geanalyseerd, wat een reeks analyseframes oplevert. Elk analyseframe beschrijft een kort stukje geluid.

Er bestaan geen methodes om een gesampled dynamisch geluidssignaal, zoals een opname van natuurlijke spraak, in zijn geheel te analyseren. Om een dynamisch geluidssignaal te analyseren, moet het worden opgedeeld in kleine eenheden waarbinnen de interessante kenmerken van het geluidssignaal min of meer constant blijven. Deze eenheden worden analysevensters (Eng: windows) genoemd. De lengte van elk analysevenster wordt vensterlengte (Eng: window length) genoemd, en is doorgaans ongeveer 10-20ms. De tijd tussen analysevensters heet de tijdsstap (Eng: time step). Om snellere veranderingen in het geluidssignaal, zoals de release van een plosief, wel in de analyse mee te kunnen nemen, wordt een tijdsstap gebruikt die korter is dan de vensterlengte. Hierdoor ontstaat overlapping tussen analysevensters. De gebruikte vensterlengte en tijdsstap hangen af van hetgeen wordt geanalyseerd en de gebruikte analysemethode. De vensterlengte en de tijdsstap zijn beide handmatig instelbaar in Praat.

Op elk analysevenster wordt een analyse uitgevoerd die de analyseframes oplevert. Op de analysevensters wordt een vensterfunctie (Eng: windowing function) toegepast, om het signaal binnen het venster niet te plotseling te laten beginnen of eindigen. Voor veel analyse-



Figuur 2.5: Een gevensterd fragment van de klinker [a].

methodes levert een analysevenster waarop geen vensterfunctie is toegepast onvoorspelbare resultaten op. De vensterfunctie heeft een maximale waarde van ten hoogste 1 in het midden van het venster en loopt af naar 0 naar het begin- en eindpunt van het venster. In figuur 2.4 is de vensterfunctie aangegeven met gekleurde curves. In figuur 2.5 is een gevensterd fragment van de klinker [a] te zien.

Elk analysevenster kan nu worden geanalyseerd met het Mel Frequency Cepstral Coefficients algoritme.

2.4.2 MFCC: Mel Frequency Cepstral Coefficients

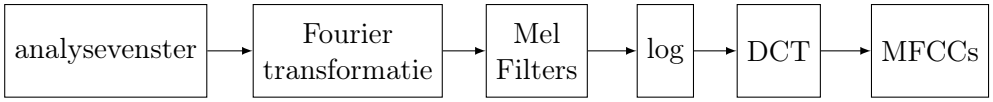
Mel Frequency Cepstral Coefficients (MFCCs) (Davis & Mermelstein, 1980) leveren een datareductiestap op, waarbij variatie in een venster die belangrijk is voor spraakperceptie behouden blijft. Het MFCC algoritme doet dit door de effecten van de bron, ofwel de F_0 zoveel mogelijk te scheiden van effecten van het filter. Ook is het mogelijk variatie in spreekvolume te negeren. Dit is bruikbaar, omdat F_0 variatie en variatie in spreekvolume weinig fonetisch interessante informatie bevatten[3]. Het scheiden van fonetisch relevante informatie en fonetisch niet-relevante informatie maakt het mogelijk de data van verschillende vensters rechtstreeks met elkaar te vergelijken.

Voor het MFCC algoritme worden de stappen uit figuur 2.6 uitgevoerd. In deze figuur is DCT een afkorting voor discrete cosinus transformatie. Elke stap uit dit algoritme wordt hieronder besproken.

Fourier transformatie

Van elk venster wordt een frequentiespectrum gemaakt. Een frequentiespectrum van een signaal geeft de energie van een frequentie aanwezig in een signaal weer. Met een frequen-

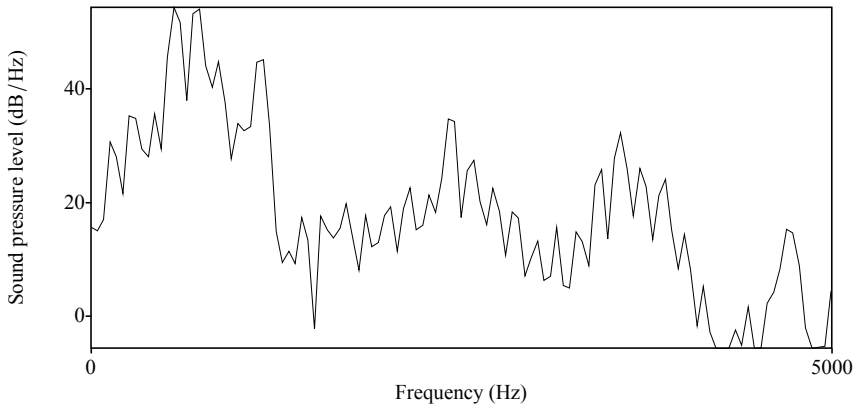
[3]Uiteraard bevat F_0 variatie wel degelijk belangrijke informatie in toontalen als het Chinees. Deze worden echter niet in de scriptie besproken



Figuur 2.6: De stappen die in het MFCC algoritme op elk analysevenster worden uitgevoerd.

tiespectrum wordt de data in een analysevenster vertaald van het tijdsdomein naar het frequentiedomein.

Voor het verkrijgen van een frequentiespectrum wordt een Fourier transformatie gebruikt. Voor een discreet geluidssignaal wordt een discrete Fourier transformatie gebruikt. In figuur 2.7 is het spectrum van het gevensterde signaal van figuur 2.5 te zien.



Figuur 2.7: Het spectrum van het gevensterde fragment van de klinker [a] uit figuur 2.5 De invloed van de F_0 is hier terug zien in de grillige vorm van het spectrum.

Mel Filters

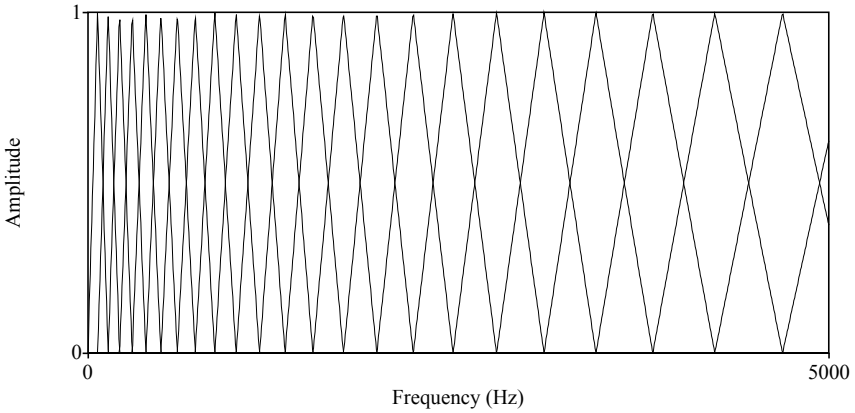
Het menselijk gehoor is niet even gevoelig voor alle frequenties. Onderzoek van onder andere Zwicker (1961), heeft aangetoond dat het menselijk gehoor gevoeliger is voor kleine frequentieverschillen bij geluiden met een lage frequentie dan bij geluiden met een hoge frequentie. Frequentieverschillen onder de 1000 Hz worden ongeveer lineair waargenomen, terwijl frequentieverschillen boven de 1000 Hz logaritmisch worden waargenomen. Er wordt daarom onderscheid gemaakt tussen frequentie van een geluid en pitch van een geluid.

Frequentie wordt gemeten in Hertz (Hz). Pitch, welke de perceptie van frequentie weergeeft, wordt gemeten in mel. Een frequentie in Hertz kan worden vertaald naar pitch in mel met formule 2.2:

$$mel(f) = 1127\ln(1 + \frac{f}{700}) \tag{2.2}$$

Voor een Mel Filterbank worden tussen de 100 en 5000 Hz driehoeksfilters toegepast

op het frequentiespectrum dat met de Fourier transformatie in de vorige stap is verkregen. Tussen de 100 en 1000 Hz worden 10 filters geplaatst, gelijkmatig verdeeld over de frequentieschaal (dus één filter op elke 100 Hz). Tussen de 1000 en 5000 Hz worden 10 filters gelijkmatig verdeeld op de mel schaal. De filters tussen de 100 en 1000 Hz staan dus elk even ver van elkaar, terwijl de filters tussen de 1000 en 5000 Hz steeds verder uit elkaar komen te staan. Voor elk driehoeksfilter geldt dat de piek van het filter, met waarde 1, ligt op de frequentie waarmee het wordt geassocieerd, en dat de zijden van de driehoek gelijkmatig aflopen naar 0. De nulpunten van een driehoeksfilter liggen op de frequenties waarmee de driehoeksfilters links en rechts van het driehoeksfilter mee worden geassocieerd. Een driehoeksfilter geplaatst op 200 Hz, tussen driehoeksfilters op 100 Hz en 300 Hz heeft dus een piek op 200 Hz, en zijden die aflopen naar 0 op 100 Hz en 300 Hz. In figuur 2.8 is een MelFilterfunctie te zien.



Figuur 2.8: Een MelFilterfunctie. De 10 melfilters tussen de 100 en 1000 Hz zijn op gelijke afstand van elkaar op de frequentieschaal geplaatst, terwijl de melfilters tussen de 1000 en 5000 Hz op gelijke afstand van elkaar op de melschaal zijn geplaatst.

Voor elk driehoeksfilter wordt nu de energie van het frequentiespectrum vermenigvuldigd met de waarde van het driehoeksfilter op dat punt. Deze waardes worden bij elkaar opgeteld, wat één waarde voor elk melfilter oplevert. Deze MelFilterwaardes kunnen echter nog niet direct worden gebruikt in het DTW algoritme, omdat broneffecten nog niet zijn gescheiden van filtereffecten. Dit gebeurt in de volgende stap.

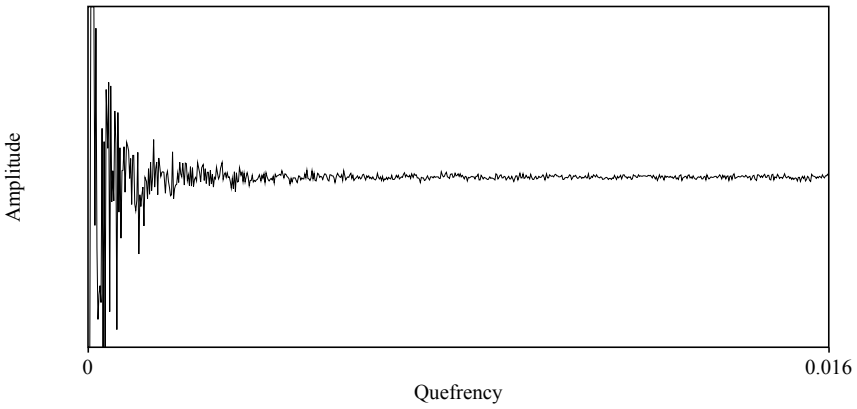
Logaritme

Van elk van de MelFilterwaardes uit de vorige stap wordt nu het logaritme genomen. De reden hiervoor is dat het menselijk gehoor gevoeliger is voor kleine amplitudeverschillen in geluiden met een kleine amplitude dan in geluiden met een grote amplitude. Door het logaritme van de MelFilter-waardes te nemen worden kleine verschillen bij kleine amplitudes versterkt ten opzichte van kleine verschillen bij grote amplitudes.

Cepstrum

Het MFCC algoritme scheidt broneffecten nu van filtereffecten door de MelFilter-waardes te behandelen als een periodiek signaal en hier een spectrum van te berekenen. Doordat een spectrum symmetrisch is, wordt in plaats van een Fourier transformatie een discrete cosinus transformatie (DCT) gebruikt. Een DCT is een Fourier transformatie waarin alleen in cosinus wordt gebruikt. Dit levert een *cepstrum* (anagram van spectrum, spreek uit /'kɛpstrʏm/) op. Het cepstrum is het spectrum van het spectrum van het geluidssignaal, en bevat cepstrale coëfficiënten in het quefrencydomein. Een quefrency is een frequentie van een oscillatie in het frequentiespectrum.

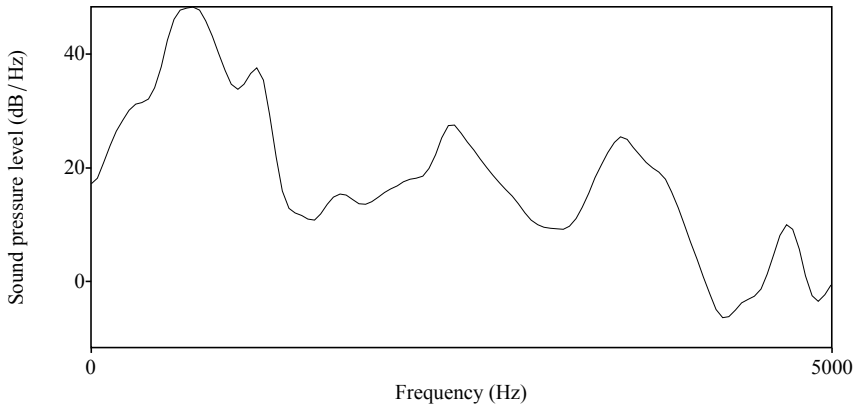
De redenering hierachter is als volgt. In het frequentiespectrum zijn pieken te zien op de F_0 en alle veelvouden hiervan. Dit is in figuur 2.7 te zien als de eerste (kleine) piek, die de F_0 aangeeft, en alle andere kleine pieken, die elk op een $n \times F_0$ afstand van de F_0 staan. Dit veroorzaakt de grillige vorm van het spectrum. Tevens zijn pieken te zien op elke hogere formant. Broneffecten en filtereffecten zijn in het cepstrum simpel van elkaar te onderscheiden. Doordat de F_0 een lagere frequentie heeft dan alle formanten, vormen de pieken veroorzaakt door de F_0 en veelvouden hiervan de snelst oscillerende component van het frequentiespectrum. In het cepstrum uit zich dit als een piek bij een hoge cepstrale coëfficiënt. Voor een hogere formant geldt dat de piek die deze veroorzaakt verder uit elkaar liggen in het frequentiespectrum, waardoor het een piek bij een lagere cepstrale coëfficiënt veroorzaakt. In figuur 2.9 is een cepstrum van gevensterde signaal in figuur 2.5 te zien.



Figuur 2.9: Het cepstrum van het spectrum uit figuur 2.7. Een kleine piek voor de F_0 is te zien midden in het cepstrum.

In het cepstrum geldt dat de laagste cepstrale coëfficiënt, c_0, informatie geeft over de gemiddelde energie in het spectrum, en sterk correleert met het spreekvolume. c_1 geeft informatie over de balans tussen lage en hoge frequenties. De hogere cepstrale coëfficiënten modelleren de steeds snellere variaties in het spectrum. Het wegfilteren van de hogere cepstrale coëfficiënten betekent dat snel oscillerende componenten uit het frequentiespec-

trum zijn gefilterd. Dit levert dus grosso modo een gladgemaakt frequentiespectrum op. In figuur 2.10 is een gladgemaakte versie van het spectrum uit figuur 2.7 te zien. Broninvloeden kunnen worden verwijderd door een hoge quefrencies uit het cepstrum te filteren.



Figuur 2.10: Een gladgemaakte versie van het spectrum uit figuur 2.7. De snelle oscilatie in het originele spectrum is hier gladgestreken, door het spectrum te regenereren uit de eerste 12 coefficienten van het cepstrum uit figuur 2.9.

Het cepstrum in figuur 2.9 is van het hele spectrum afgebeeld in figuur 2.6 genomen, niet de MelFilterwaardes. Het cepstrum bevat hierdoor evenveel waardes als het spectrum. Voor MFCC's worden echter de MelFilterwaardes gebruikt voor het berekenen van een cepstrum. Als, net als in afbeelding 2.8, 23 MelFilterwaarders worden gebruikt, levert een DCT op deze waardes 23 cepstrale coëfficiënten op. Door de hogere cepstrale coëfficiënten weg te filteren, worden broninvloeden uit het cepstrum gefilterd. In spraaktoepassingen worden doorgaans de eerste 12 van deze coëfficiënten gebruikt. Een analyseframe van een analysevenster bevat nu een vector met deze 12 coëfficiënten. Deze vectoren worden feature vectors genoemd. Een signaal waaruit I analysevensters kunnen worden geëxtraheerd, levert dus I feature vectors op.

2.4.3 Algoritme

Voor zowel het inputsignaal als het voorbeeldsignaal wordt de bovenstaande analysestap uitgevoerd. Dit levert I featurevectors $\mathbf{x}_{1..I}$ op voor het inputsignaal, en J featurevectors $\mathbf{y}_{1..J}$ voor het voorbeeldsignaal. In de notatie van deze sectie wordt met $\mathbf{x}_i$ de i-de vector van het inputsignaal bedoeld, waarbij $1 \leq i \leq I$. Elke feature vector bevat 12 cepstrale coëfficiënten: $\mathbf{x}_{i_{1..12}}$.

Het DTW algoritme lijnt twee geluidssignalen op door de series feature vectors van de respectievelijke geluidssignalen met elkaar op te lijnen. Dit gebeurt, net als in het string alignment algoritme, door de kortste bewerkingsafstand tussen de twee series vectoren te bepalen. Het bronsignaal is hier het inputsignaal, en het doelsignaal het voorbeeldsignaal.

Een belangrijk verschil met het string alignment algoritme is dat lettertekens een beperkte set vormen. De feature vectors bevatten echter reële getallen, waardoor er een oneindig aantal mogelijke feature vectors bestaat. Substitutiekosten worden hierom berekend met behulp van de Euclidische afstand tussen twee feature vectors:

$$d_{i,j} = \| \mathbf{x}_i - \mathbf{y}_j \| = \sqrt{\sum_{k=1}^{12} (\mathbf{x}_{i_k} - \mathbf{y}_{j_k})^2} \tag{2.3}$$

In figuur 2.11 is een visualisatie van een matrix met de euclidische afstand tussen elk paar vectoren $(\mathbf{x}_i, \mathbf{y}_j)$ te zien. Index k is de k-de cepstrale coëfficiënt.

Nu kan een cumulatieve afstandstandsmatrix D, vergelijkbaar met de bewerkingsafstandsmatrix voor het string alignment algoritme, worden opgesteld. Dit gebeurt op dezelfde wijze als voor het string alignmentalgoritme. Eerst wordt een $(I+1) \times (J+1)$ matrix opgesteld. Element 0,0 krijgt wederom de waarde 0. Vanaf daar wordt de waarde van elk element van $D_{i,j}$ als volgt bepaald:

$$D_{i,j} = \min \begin{pmatrix} D_{i-1,j} + d_{i,j}, \\ D_{i-1,j-1} + 2d_{i,j}, \\ D_{i,j-1} + d_{i,j} \end{pmatrix} \tag{2.4}$$

In deze formule is te zien dat nu, in tegenstelling tot het string alignment algoritme, voor elke bewerking de afstand tussen elementen meetelt. Substitutiekosten worden vervangen door $2d_{i,j}$. De vermenigvuldiging met 2 vindt plaats om een schuine stap in de matrix geen oneerlijk voordeel te geven boven een horizontale en verticale stap.

De beperkingen die gegeven werden voor het string alignment algoritme vereisen voor het DTW algoritme enige aanpassing. Ten eerste geldt de derde beperking – het pad moet vanaf het begin van beide series feature vectors het eind van beide series feature vectors lopen – niet meer zo sterk. De beperking wordt voor het DTW algoritme als volgt aangepast:

3. De begin- en eindpunten van het pad binnen op een zekere afstand van respectievelijk het begin en het eind van beide series feature vectors liggen.

Daarnaast wordt een vierde toegevoegd.

4. Het pad moet aan een hellingsbeperking voldoen. Dit is om te voorkomen dat een pad op langere stukken volledig horizontaal of verticaal loopt. Een diagonale stap betekent dat het bronsignalen voor dat analyseframe met het doelsignaal matchet. Een verticale stap betekent een deletie van dit analyseframe uit het bronsignaal. Een horizontale stap betekent een insertie van het analyseframe in het doelsignaal. Als het pad lange verticale stukken bevat, betekent dit dat een lang deel van het bronsignaal wordt verwijderd.

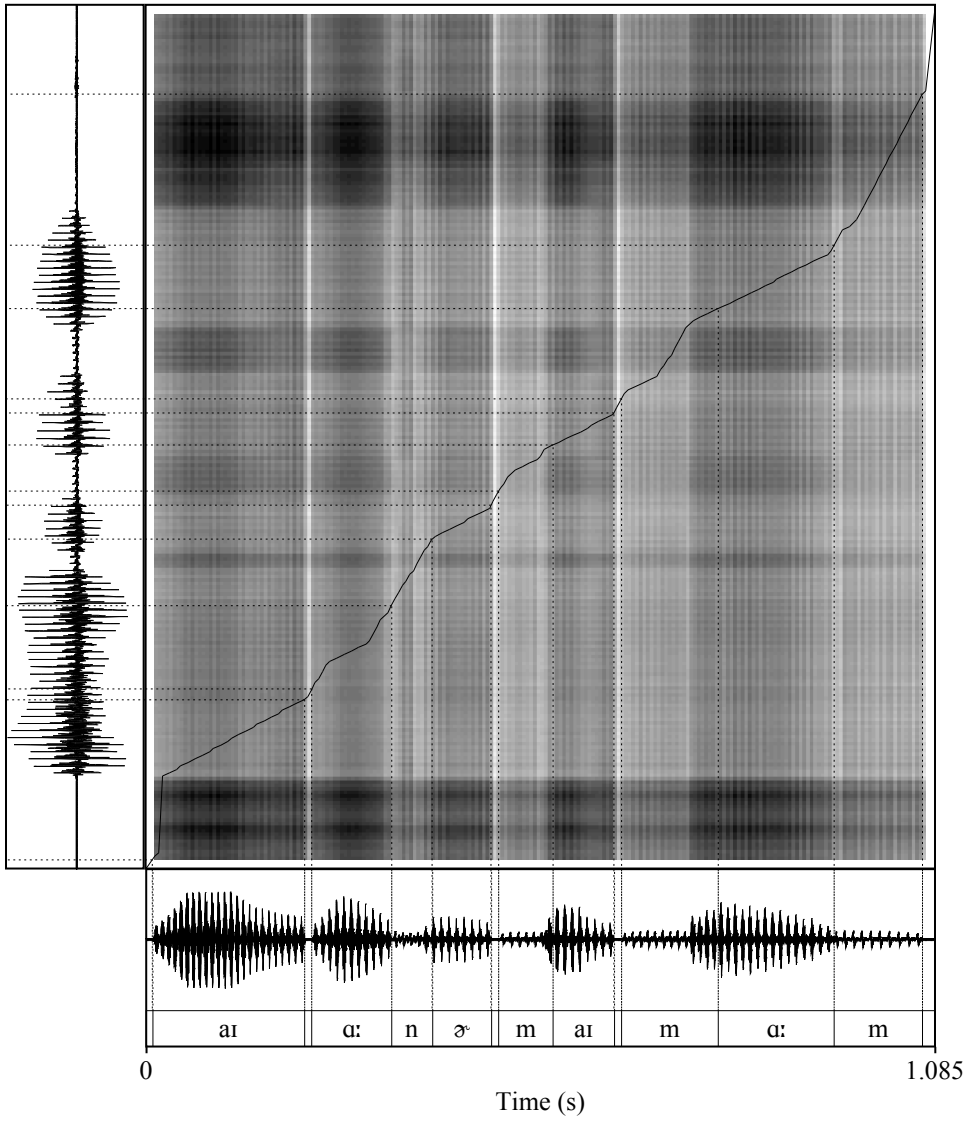
Een hellingsbeperking stelt hoeveel horizontale of verticale stappen mogen worden gezet, voordat een diagonale stap moet worden gezet. Een hellingsbepering van 2/3-3/2 bijvoorbeeld bepaalt dat voor elke twee horizontale of verticale stappen, drie diagonale stappen moeten worden gezet. In figuur 2.12 is een cumulatieve afstandsmatrix te zien, waarin een hellingsbeperking van 3/2-2/3 wordt gebruikt. De polygoon geeft aan welke punten in de matrix gegeven de hellingsbeperking kunnen worden bereikt. De zwarte lijn door de matrix geeft het kortste pad aan.

De hellingsbeperking impliceert dat er een maximale lengteverhouding tussen twee signalen mag bestaan. Als het lengteverschil tussen de signalen groter is dan de hellingsbeperking, heeft het geen zin om DTW uit te voeren op de signalen, omdat er dan geen mogelijk pad door de afstandsmatrix kan bestaan dat overal aan de hellingsbeperking voldoet.

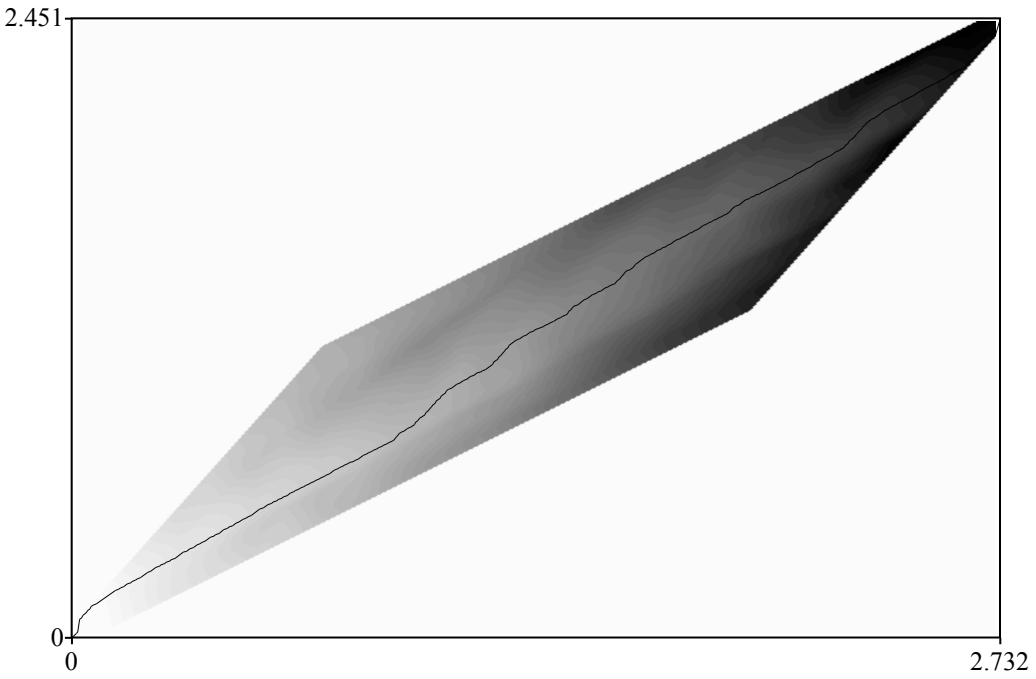
Het kortste pad door de matrix, en daarmee de oplijning van de signalen, vindt op dezelfde wijze als bij het string alignment algoritme plaats.

2.4.4 Toepassing van het TTS-DTW systeem in Praat

Als een geannoteerd voorbeeldsignaal is gemaakt, en het DTW algoritme is uitgevoerd op het voorbeeldsignaal en het inputsignaal, kan de vertaling van de annotatie van het voorbeeldsignaal met de annotatie van het inputsignaal worden uitgevoerd. Dit gebeurt door alle begin- en eindpunten van segmenten in de annotatie van het voorbeeldsignaal te vertalen naar overeenkomstige tijdspunten in het inputsignaal. Dit gebeurt aan de hand van de oplijning van de twee signalen die door het DTW algoritme is gevonden. Dit proces is te zien in afbeelding 2.11.



Figuur 2.11: De een matrix met alle afstanden $d_{i,j}$ voor twee geluidssignalen. De middelste zwarte lijn geeft het pad van de kortste bewerkingsafstand aan. De feature vectors op de verticale as komen van het bronsignaal, de feature vectors op de horizontale as van het gesynthetiseerde doelsignaal. Lichtere punten betekenen een kleinere euclidische afstand tussen twee feature vectors. De gestippelde lijnen geven aan hoe grenzen tussen fonemen in het voorbeeldsignaal worden overgezet naar het inputsignaal. De hier uitgesproken zin is 'I honor my mom.'



Figuur 2.12: De cumulatieve afstandsmatrix met een hellingsbeperking van 3/2-2/3. Een donkere kleur geeft een grotere cumulatieve afstand aan. De zwarte lijn geeft het korste pad aan. Voor elk punt op de lijn geldt dat het pad daarnaar toe de kleinste cumulatieve afstand tot dan toe heeft.

Hoofdstuk 3

Werkwijze

De werking van het TTS-DTW systeem in Praat zal worden geëvalueerd aan de hand van een vergelijking met de handmatige segmentering van bestaande corpora. Dit zal gebeuren voor een Engelstalig corpus: het TIMIT corpus (Lamel et al., 1986), en een Nederlandstalig corpus: het IFA corpus (van Son et al., 2001).

3.1 Annotatieconventies spraaksynthesizer

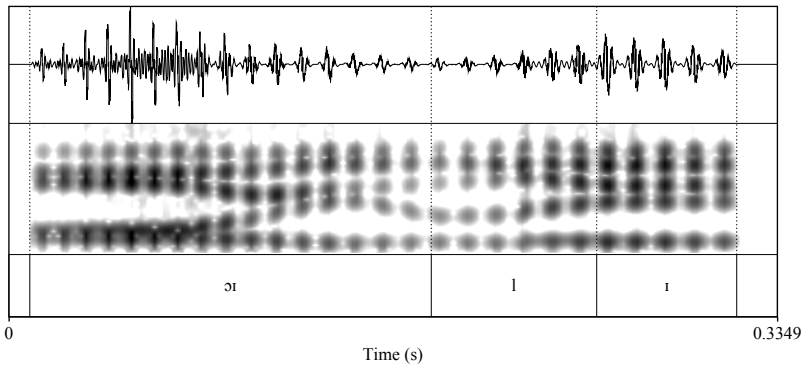
Zoals besproken in sectie 2.1 kunnen de keuzes die gemaakt zijn in de annotatie van corpora verschillen. Hierdoor komen grenzen tussen fonemen niet op vergelijkbare plekken te liggen. Om een vergelijking te kunnen maken met de annotatie van de spraaksynthesizer, en dus de annotatie die in het TTS-DTW systeem naar het inputsignaal wordt vertaald, en de handmatige annotatie van de corpora, moeten de annotaties van de corpora worden omgezet naar een annotatie die overeenkomt met de annotatie van de spraaksynthesizer.

Hiertoe moet eerst bekend zijn hoe de spraaksynthesizer het voorbeeldsignaal annoteert. Uit de uitspraakregels en het uitspraakwoordenboek kan globaal worden bepaald wat de annotatie van een signaal zal zijn. Uit deze bestanden wordt duidelijk tussen welke fonemen eSpeak een grens zal plaatsen, maar niet waar deze grens exact zal vallen. Daarom moet handmatig elke foonovergang worden bekeken.

De spraaksynthesizer maakt gebruik van de Kirshenbaum (Kirshenbaum, 2001) transcriptie. Zoals eerder besproken maakt de Kirshenbaum transcriptie gebruik van ASCII symbolen om IPA symbolen weer te geven.

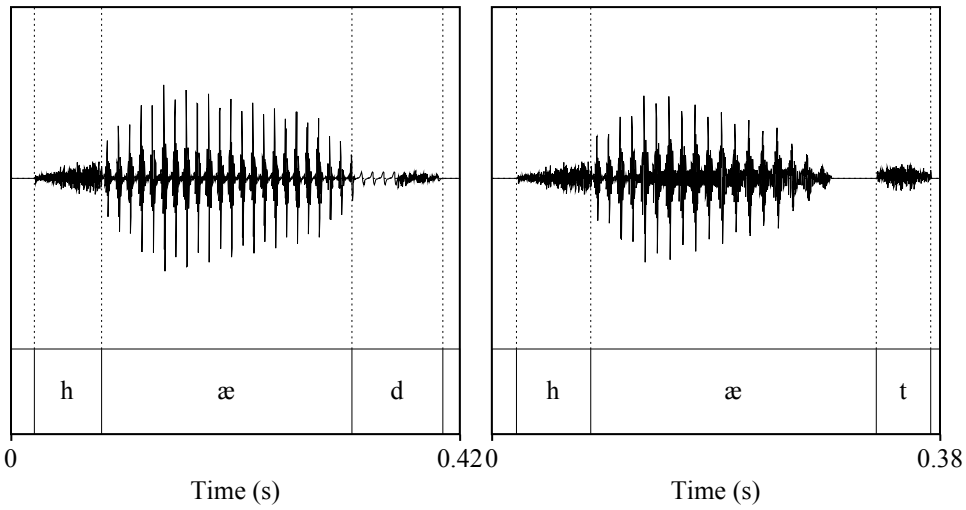
Een grens tussen twee fonen $f_1 \rightarrow f_2$ wordt door de spraaksynthesizer getrokken op het moment dat f_2 stabiel is. Overgangsverschijnselen van f_1 naar f_2 worden geannoteerd alsof deze bij f_1 horen. Hierdoor zijn geannoteerde fonen nooit besmet door het voorgaande foneem, maar is in elke geannoteerde foon de invloed van de volgende foon sterk aanwezig. Overigens vertonen fonen in de spraaksynthesizer een langere stabiele fase dan fonen in continue natuurlijke spraak, omdat de spraaksynthesizer gebruik maakt van formantsynthese.

In figuur 3.1 is een voorbeeld hiervan te zien.



Figuur 3.1: De eigen annotatie van de spraaksynthesizer voor het woord 'oily'. De grens tussen het [l] segment en het [i] segment wordt pas geplaatst op het moment dat de formanttransitie van [l] naar [i] voltooid is.

De spraaksynthesizer gebruikt een opmerkelijke annotatie bij plosieven. In het geval van een stemloze plosief wordt de korte stilte voor de release die in natuurlijke spraak gepaard gaat met het afsluiten van het spraakkanaal als onderdeel van de vorige spraakklank geannoteerd. Als het een stemhebbende plosief betreft, bevat de sluiting vocal murmur en wordt deze als onderdeel van de plosief geannoteerd (zie figuur 3.2).



Figuur 3.2: De spraaksynthesizerannotatie van een stemhebbende plosief ([d], figuur links) en een stemloze plosief ([t], figuur rechts). Het verschil in voice onset time tussen de [d] en de [t] veroorzaakt een verschil in annotatie: de sluiting van de [d] wordt als onderdeel van de [d] geannoteerd, en de sluiting van de [t] als onderdeel van het voorgaande segment.

3.2 Gebruikte corpora

3.2.1 Het TIMIT corpus

Het TIMIT corpus (Lamel et al., 1986) is een corpus van gesproken Amerikaans-Engels. Het corpus bevat gesproken zinnen van 630 sprekers, afkomstig uit acht verschillende dialectregio's.

De opgenomen data bestaat uit twee shibboleth-zinnen: zinnen die fonemen bevatten waarvan de uitspraak merkbaar verschilt per dialectregio. Beide shibboleth-zinnen zijn door elke spreker zijn opgenomen:

1. She had your dark suit in dirty wash water all year.
2. Don't ask me to carry an oily rag like that.

Daarnaast zijn 450 fonetisch rijke zinnen opgenomen. Voor elke spreker zijn 5 van deze zinnen opgenomen, en elke van de 450 zinnen is door 7 verschillende sprekers opgenomen. Deze zinnen zijn ontworpen om een zo groot mogelijk aantal interessante difonen te bevatten. Een voorbeeld van een fonetisch rijke zin:

> The fog prevented them from arriving on time.

Ten slotte bevat het corpus 1,890 fonetisch diverse zinnen, elk opgenomen door één spreker. Elke spreker heeft drie van deze zinnen opgenomen. Deze zinnen zijn afkomstig uit de Brown en Playwrights Dialog corpora. De zinnen zijn geselecteerd om de variatie in allofonische context te maximaliseren:

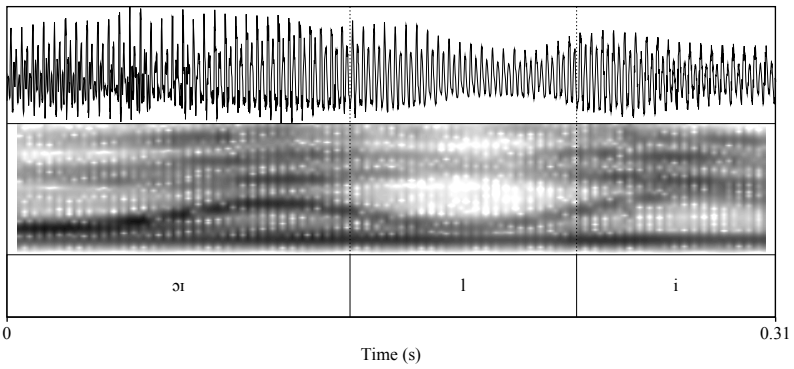
> For once radicalism was a recommendation.

In totaal zijn door elke spreker dus 10 zinnen opgenomen, wat een corpus oplevert dat 6,300 zinnen bevat.

Annotatieconventies

Grenzen tussen fonen zijn in TIMIT zodanig getrokken dat wederzijdse besmetting tussen fonemen minimaal is. Bijvoorbeeld in de overgang van /l/ naar /i/ in het woord 'oily' in de tweede shibboleth-zin (zie hierboven) wordt de grens getrokken op het punt waar de afstand tussen de stabiele fase van de beide fonemen het grootst is. In figuur 3.3 staat hiervan een voorbeeld.

De transcriptie van het TIMIT corpus verschilt sterk van de transcriptie die de spraaksynthesizer levert. In het TIMIT corpus wordt onderscheid gemaakt tussen een groter aantal fonemen. Bovendien wordt de sluiting van een plosief los van de release geannoteerd. Verder is de combinatie klinker + liquide in het TIMIT corpus apart gesegmenteerd, terwijl dit door de spraaksynthesizer als één foneem wordt geannoteerd.



Figuur 3.3: De handmatige annotatie in het TIMIT corpus van het woord 'oily', afkomstig uit TIMIT-zin sa2, uitgesproken door spreker faks0. De grens tussen /l/ en /i/ wordt getrokken op het punt waar de afstand tot de stabiele fase van de twee fonemen het grootst is.

Om de annotatie van het TIMIT corpus te kunnen vergelijken met die van de spraaksynthesizer, is een klein Python script geschreven dat de annotatie van het TIMIT corpus omzet naar de annotatieconventies van de spraaksynthesizer. In Appendix A: Annotatie-aanpassingen wordt het verschil tussen de segmentering en annotatie van TIMIT en de spraaksynthesizer uitgebreider besproken. Het is niet mogelijk om de grens tussen fonen in TIMIT zo aan te passen dat deze op een plek vergelijkbaar met die van de spraaksynthesizer komt te liggen. Hierdoor zal voor bepaalde foneemovergangen de grens door de spraaksynthesizer later worden gelegd dan in TIMIT het geval is. Hier wordt in de resultaten verder op ingegaan.

Het TIMIT corpus is getranscribeerd in het Arpabet. Evenals de Kirshenbaum transcriptie is Arpabet een transcriptie van Engelse spraakklanken in ASCII symbolen. De omzetting van de TIMIT annotatie naar een annotatie die compatibel is met die van de spraaksynthesizer is beschreven in Appendix A: Annotatie-aanpassingen.

3.2.2 Het IFA corpus

Het IFA corpus is een Nederlandstalig spraakcorpus. Het corpus bevat opgenomen spraak van acht moedertaalsprekers van het Nederlands: vier mannen en vier vrouwen. De sprekers variëren in leeftijd van 15 tot 66 jaar.

Van elke spreker is spraak opgenomen in acht situaties:

1. Het vertellen van een informeel verhaal aan een 'interviewer'
2. Het uit het hoofd navertellen van een van te voren gelezen verhaaltje.
3. Het oplezen van een verhaaltje van blad
4. Het oplezen van een willekeurige lijst van alle zinnen uit de verhaaltjes

5. Het oplezen van pseudo-zinnen, waarin elk woord is vervangen door een woord van dezelfde woordsoort

6. Het oplezen van unieke syllaben- en woordenlijsten.

7. Het oplezen van idiomatische sequenties (het alphabet en de nummers 1 t/m 12) en diagnostische sequenties (losse klinkers, /hVd/ en /CVC/ lijsten).

Situaties 3, 4 en 5 komen overeen met de manier waarop de spraakdata in TIMIT is geëliciteerd. Alleen deze situaties worden gebruikt in de evaluatie van het algoritme.

Annotatieconventies

Het IFA corpus is getranscribeerd in SAMPA. Net als het Arpabet en de Kirshenbaum-transcriptie is het SAMPA ontworpen om fonetische transcriptie met enkel ASCII symbolen mogelijk te maken.

De annotatieconventies van het IFA corpus worden het best samengevat in het onderstaande citaat uit de handleiding annotatieconventies IFA, vetgedrukt door MvB:

> "Bij het oplijnen van de foneemlabels en hun grenzen met de spraak worden de grenzen gezet op plaatsen waar de golfvorm laat zien dat er iets veranderd [sic] aan de spraak. Welke verandering in de golfvorm de grens aangeeft wordt bepaald door de structuur van de golfvorm, de CoG [Center of Gravity, de spectrale balans - MvB] of het gehoor. **Het doel is altijd de foneemgrenzen zo te plaatsen dat ieder foneemsegment zo zuiver mogelijk is, terwijl buurfonemen zo min mogelijk 'besmet' zijn**."

Hier wordt expliciet gekozen voor een annotatie die besmetting minimaliseert, waarmee de annotatieconventies van het IFA corpus overeenkomen met die van het TIMIT corpus. Ook hier verschillen de annotatieconventies van het corpus van die van de spraaksynthesizer. De effecten hiervan worden in de resultaten besproken.

De SAMPA transcriptie komt grotendeels overeen met de Kirshenbaum transcriptie. De omzetting van de IFA annotatie naar een annotatie die compatibel is met die van de spraaksynthesizer is beschreven in Appendix A: Annotatie-aanpassingen.

3.3 Vergaring Resultaten

In beide corpora wordt het TTS-DTW algoritme uitgevoerd op elke geannoteerde uiting. Als input van het algoritme worden het geluidsbestand behorende bij een uiting en een woordelijke transcriptie van die uiting genomen. De output is een automatische annotatie.

De spraaksynthesizer is ingesteld op een mannelijke standaard Amerikaans-Engelse stem voor het TIMIT corpus, en een mannelijke Nederlandse stem voor het IFA corpus. De spraaksynthesizer is ingesteld om geen stilte tussen woorden te laten vallen, en een minimale variatie in pitch te laten zien.

Het TTS-DTW systeem biedt de mogelijkheid de spreeksnelheid aan te passen aan het inputsignaal, door op basis van het aantal woorden in het inputsignaal en de lengte van het inputsignaal een schatting van het spreektempo te maken. Om te testen of dit een nauwkeurigere annotatie oplevert, wordt bovenstaande procedure herhaald met correctie op spreeksnelheid.

Daarnaast is het mogelijk om enkel een foneemstring als inputtekst te gebruiken. Hiermee kunnen de effecten van een foute tekst-naar-foneem omzetting worden omzeild. Hiertoe wordt de hierboven beschreven procedure nog eenmaal uitgevoerd, met een fonetische transcriptie als inputtekst. De foneemstring wordt gevormd door de losse fonemen uit de handmatige annotaties samen te voegen tot één reeks fonemen. Woordgrenzen worden hierin niet aangegeven. Hier is voor gekozen, omdat ook tussen woorden coarticulatieverschijnselen optreden. Hierdoor is in veel gevallen niet vast te stellen of een foneem op een woordgrens bij het voorgaande of het volgende woord hoort. Bovendien is door de omzettingen van annotaties, zoals beschreven in Appendix A, niet voor elk segment vast te stellen tot welk woord het behoort.

De kwaliteit van de automatische annotatie wordt bepaald aan de hand van een vergelijking met de handmatige annotatie van elke uiting. Zoals eerder beschreven kan de automatische annotatie een ander aantal segmenten bevatten dan de handmatige. Om bij elkaar behorende segmenten te vinden, wordt het eerder beschreven string alignment algoritme uitgevoerd op de handmatige annotatie met de automatische annotatie. De voor de string alignment gebruikte substitutiekostenmatrices (één voor het TIMIT corpus en één voor het IFA corpus) zijn opgenomen in Appendix B: Substitutiekostenmatrices.

3.4 Analyse resultaten

3.4.1 Matchende segmenten

Het string alignmentalgoritme maakt een tabel die de alignment bevat. In tabel 3.1 staat een fragment van een alignmenttabel voor een uiting uit TIMIT. In deze tabel worden grenzen tussen segmenten in de handmatige annotatie gezocht. Alleen de grenzen tussen segmenten uit de handmatige annotatie waarvoor matchende segmenten uit de automatische annotatie kunnen worden gevonden, worden meegenomen. Dit betekent dat als tussen twee segmenten in de handmatige annotatie een segment is ingevoegd in de automatische annotatie, of als een segment uit de handmatige annotatie is verwijderd ten opzichte van de automatische annotatie, de grens tussen deze segmenten niet wordt geteld. De reden hiervoor is dat

Tabel 3.1: Een fragment van een String Alignment tabel. De source data heeft betrekking op fonemen in de handmatige annotatie, in dit geval TIMIT, en de target data verwijst naar de automatische annotatie. In de automatische annotatie is een extra stilte aanwezig in het begin, en worden voor twee fonemen een substitutie uitgevoerd. In de operation-kolom geeft 'i' een insertie aan, 'd' een deletie en 's' een substitutie.

sourceText	sourceStart	sourceEnd	targetText	targetStart	targetEnd	operation
?	–undefined–	–undefined–	_	0	0.542	i
_	0	0.131	_	0.542	0.579	
p	0.131	0.157	p	0.579	0.614	
ʊ	0.157	0.213	ʊ	0.614	0.707	
_	0.213	0.273	t	0.707	0.738	s
ð	0.273	0.303	ð	0.738	0.790	
ʌ	0.303	0.345	ə	0.790	0.836	s
b	0.345	0.438	b	0.836	0.905	
ʊ	0.438	0.543	ʊ	0.905	0.986	
ʧ	0.543	0.608	ʧ	0.986	1.042	

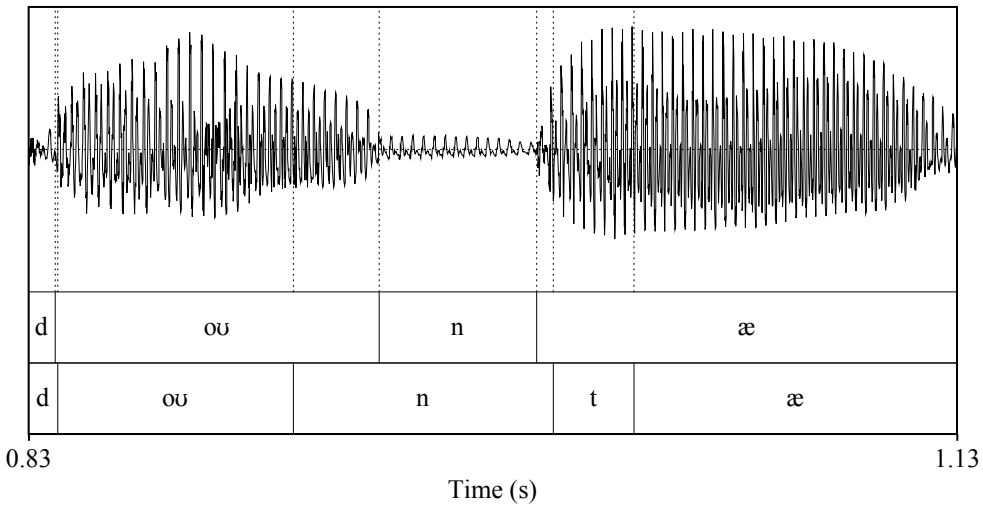
er in het geval van verwijderde of ingevoegde segmenten niet met zekerheid een grens in de automatische annotatie is vast te stellen die matcht met een grens uit de handmatige annotatie. Deleties en inserties van segmenten hebben uiteraard wel invloed op de kwaliteit van de annotatie. Hoe met deleties en inserties wordt omgegaan wordt behandeld in de volgende secties.

Van elke gevonden grens wordt bepaald hoe groot de afwijking van de grens in de automatische annotatie is ten opzichte van de grens in de handmatige annotatie.

3.4.2 Inserties

Inserties van segmenten in de automatische annotatie waarvoor geen overeenkomstige spraakklank in het inputsignaal is te vinden, zijn een veelvoorkomend verschijnsel. Figuur 3.4, een afbeelding van een fragment zin sa2 uit het TIMIT corpus, uitgesproken door spreker faks0, bevat een voorbeeld van insertie. Het effect van inserties wordt onderzocht, door voor elke grens in de handmatige annotatie waar een insertie heeft plaatsgevonden, de segmenten in de automatische annotatie te vinden die matchen met de segmenten waartussen een insertie heeft plaatsgevonden. Als deze segmenten grenzen aan het geïnserteerde segment, wordt de eindgrens van het segment voor het geïnserteerde segment, en de begingrens van het segment na het geïnserteerde segment vergeleken met de grens uit de handmatige annotatie waartussen het geïnserteerde segment is geplaatst. Deze procedure kan worden uitgebreid naar meerdere, aansluitend geïnserteerde segmenten. Alleen grenzen van segmenten die direct naast één of meerdere geïnserteerde segmenten liggen en matchen met de segmenten in de handmatige annotatie worden meegenomen in de evaluatie. In het geval van een deletie en een insertie op dezelfde locatie is niets zinnigs meer te zeggen over de afwijking van de grenzen.

In figuur 3.4 zouden de eindes van de [n] segmenten en de onsets van de [æ] segmenten



Figuur 3.4: Een insertie in een inputsignaal van TIMIT. De bovenste annotatie is de handmatige annotatie, de onderste annotatie is de automatische annotatie. Het [t]-segment is geïnserteerd ten opzichte van de annotatie van TIMIT.

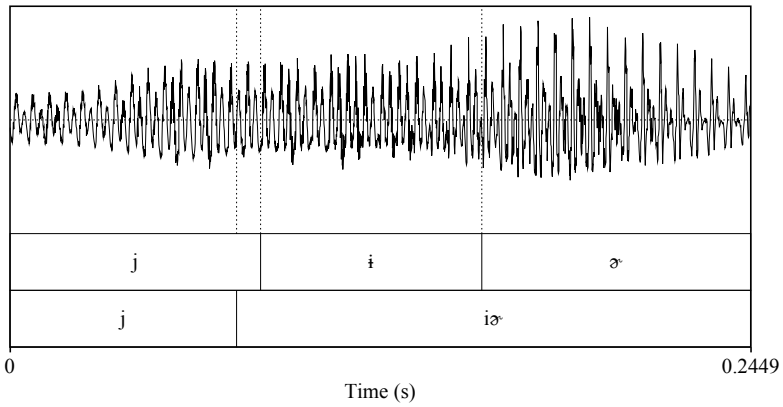
met elkaar worden vergeleken.

3.4.3 Deleties

In figuur 3.5 staat een voorbeeld van een deletie. Deleties vormen een klein deel van de wijzigingen ten opzichte van de bronannotatie. Deleties komen voornamelijk voor in het geval van twee fonemen die apart zijn geannoteerd in de handmatige annotatie, maar één foneem vormen in de annotatie van de spraaksynthesizer. Analyse van deleties op dezelfde wijze als analyse van inserties zal daarom geen interessante informatie opleveren.

3.5 Evaluatie Resultaten

In de data zal eerst worden gezocht naar invloeden van dialect en spreker. Daarna zal worden onderzocht welke afwijkingen worden gevonden op grenzen tussen matchende segmenten. Verder zal worden bekeken welke invloed inserties en deleties op de automatische annotatie van spraak hebben.



Figuur 3.5: Een deletie uit een bronannotatie van TIMIT. De bovenste annotatie is de handmatige annotatie, de onderste annotatie is de automatische annotatie. Het [ɨ]-segment is verwijderd ten opzichte van de annotatie van TIMIT.

Hoofdstuk 4

Resultaten

In de tekst hieronder wordt naar drie typen methodes van annotatie verwezen: annotatie met woorden als input zonder correctie op spreeksnelheid, annotatie met woorden als input met correctie op spreeksnelheid en annotatie met een foneemstring als input. Deze zullen in de tekst worden aangeduid als respectievelijk de aligned methode, de aligned_corr methode en de aligned_phon methode.

De statistische tests zijn alle uitgevoerd met de ingebouwde statistische functies van Praat (mediaan en MAD) en de standaard "Stats" library van R (R Development Core Team, 2012) (alle overige statistische toetsen).

4.1 Problemen met de output van het TTS-DTW algoritme in Praat

4.1.1 Onannoteerbare bestanden

Op een aantal bestanden kon het TTS-DTW algoritme niet worden uitgevoerd. Doordat PraatScript geen manier heeft om errors op te vangen en te verwerken, kan alleen worden gerapporteerd dat annotatie niet is gelukt voor deze bestanden. Een mogelijke oorzaak voor een aantal bestanden is dat de lengte van het inputsignaal teveel verschilde van de lengte van het voorbeeldsignaal, waardoor de hellingsbeperking van het DTW algoritme werd overschreden. In tabel 4.1 is voor elk corpus te zien op welk percentage bestanden dit in elke methode van toepassing was.

4.1.2 Onbruikbare annotaties

In een aantal gevallen kan de output van het TTS-DTW algoritme niet meer door Praat worden gelezen. Dit kwam alleen voor bij automatische annotatie op het Nederlandstalige IFA corpus. De oorzaken hiervan zijn nog niet bekend.

Tabel 4.1: Het percentage bestanden dat niet kon worden geannoteerd door het TTS-DTW algoritme. TIMIT bevat in totaal 6,300 geannoteerde uitingen, IFA bevat 4,454 geannoteerde uitingen. Deze tabel heeft, in tegenstelling tot de rest van dit hoofdstuk betrekking op het gehele IFA corpus.

corpus	**methode**	**Aantal onannoteerbare bestanden**
TIMIT	aligned	1.07%
	aligned_corr	1.54%
	aligned_phn	0.60%
IFA	aligned	2.56%
	aligned_corr	1.35%

4.2 Totaalresultaten corpora

4.2.1 Inserties, deleties en substituties

In deze sectie wordt besproken hoeveel inserties, deleties en substituties hebben plaatsgevonden bij het automatisch annoteren van de corpora.[1] Dit geeft een grove maat voor de prestaties van de text-naar-fonemen stap van de spraaksynthesizer. De resultaten zijn opgesplitst naar corpus en annotatiemethode. In tabel 4.2 staat per corpus per annotatiemethode het aantal inserties, deleties en substituties.

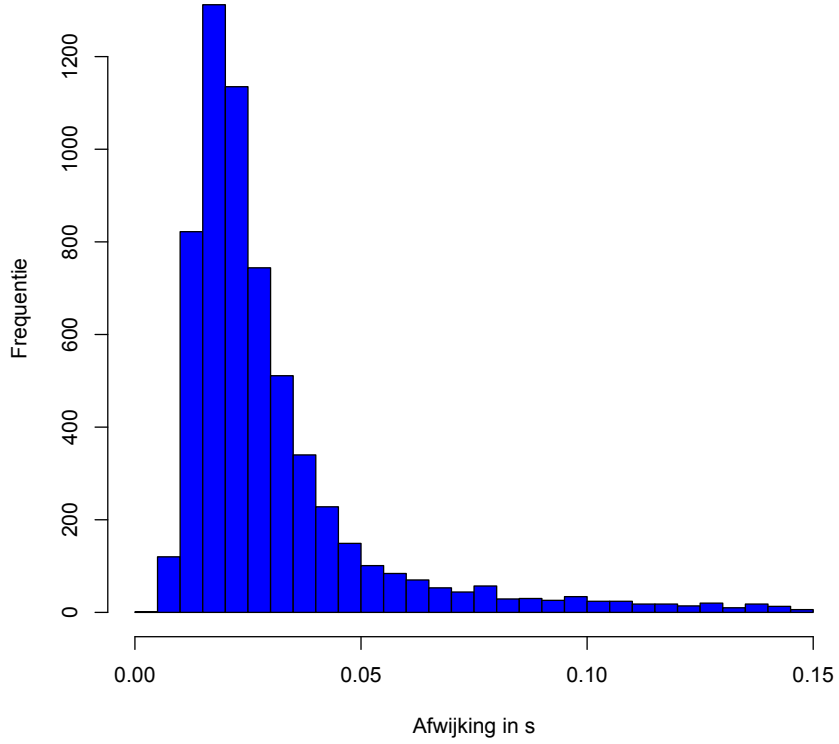
Tabel 4.2: Het aantal inserties, deleties en substituties per corpus, per annotatiemethode. Het percentage inserties is een percentage van alle automatisch geannoteerde segmenten, de percentages deleties en substituties zijn percentages van alle handmatig geannoteerde segmenten. De tot_h en tot_a kolommen bevatten respectievelijk het aantal segmenten in de handmatige annotatie en het aantal segmenten in de automatische annotatie.

corpus	**methode**	**inserties**	**deleties**	**substituties**	**tot_h**	**tot_a**
TIMIT	aligned	10.01%	2.69%	22.33%	205,010	228,582
	aligned_corr	10.01%	3.00%	22.21%	204,300	220,404
	aligned_phn	5.43%	1.36%	1.66%	208,715	217,707
IFA	aligned	12.51%	6.06%	9.50%	38,394	41,516
	aligned_corr	12.58%	6.13%	10.33%	37,661	40,453

4.2.2 Vergelijking annotatiemethodes

In deze sectie worden de totaalresultaten van de corpora met elkaar vergeleken. Om hier resultaten te verkrijgen, wordt de absolute afwijking per grens tussen fonemen ten opzichte van de handmatige annotatie gebruikt. De datapunten verkregen op de manier beschreven in het vorige hoofdstuk zijn echter binnen annotatiebestanden niet onafhankelijk: als één grens in de automatische annotatie sterk is verschoven ten opzichte van de handmatige annotatie, worden hierdoor andere grenzen ook 'weggeduwd'. Om toch onafhankelijke datapunten

[1] Vanwege een probleem met het verwerken van lange foneemstrings door de spraaksynthesizer, zijn voor het IFA corpus de resultaten waarbij de foneemstring als invoer werd gebruikt niet opgenomen in deze scriptie. Verder is, zoals besproken in het vorige hoofdstuk, alleen een subset van het IFA corpus gebruikt.



Figuur 4.1: Een histogram van de verdeling van de afwijkingen ten opzichte van TIMIT, automatisch geannoteerd met de aligned methode. De afwijkingen die zijn gebruikt voor het maken van deze histogram zijn medianen van de afwijkingen van elke opgenomen uiting.

te kunnen gebruiken, zijn de medianen van de absolute afwijkingen per annotatiebestand gebruikt als datapunten.

De medianen per bestand zijn niet symmetrisch verdeeld. Dit is te zien in figuur 4.1. waar een histogram van de verdeling van afwijkingen van alle medianen per bestand van TIMIT met de aligned methode staan.

Dit levert per corpus per annotatiemethode de volgende resultaten op:

Tabel 4.3: De mediaan en MAD van de medianen van elk bestand, per corpus per annotatiemethode. Alle data is secondes.

corpus	**methode**	**Mediaan**	**MAD**
TIMIT	aligned	0.024	0.011
	aligned_corr	0.028	0.015
	aligned_phn	0.028	0.016
IFA	aligned	0.025	0.010
	aligned_corr	0.029	0.013

Met de Cramer-von Mises test is nagegaan of de data normaal verdeeld is. Dit bleek niet het geval. Hierdoor kon een ANOVA niet worden gebruikt. In plaats daarvan is gebruik gemaakt van de Mann-Whitney U test. Deze is uitgevoerd op de twee methodes in IFA, en op elk paar van de drie methodes in TIMIT. Om te voorkomen dat in het laatste geval problemen met herhaalde tests ontstaan, is een Bonferroni correctie uitgevoerd. Omdat

3 tests zijn uitgevoerd moet de maximale p-waarde die nog significant is door 3 worden gedeeld. Dit betekent dat p-waarden kleiner dan 0.0167 significant zijn.

Hieruit blijkt dat in IFA de resultaten van de aligned_corr methode significant sterker afwijken dan de resultaten van de aligned methode ($p<0.001$). Verder doen de aligned_corr en de aligned_phn methodes in TIMIT het significant slechter dan de aligned methode ($p<0.001$), hoewel de aligned_corr en aligned_phn methodes niet significant van elkaar afwijken.

4.3 Verschillen tussen groepen in de corpora

4.3.1 Het TIMIT corpus

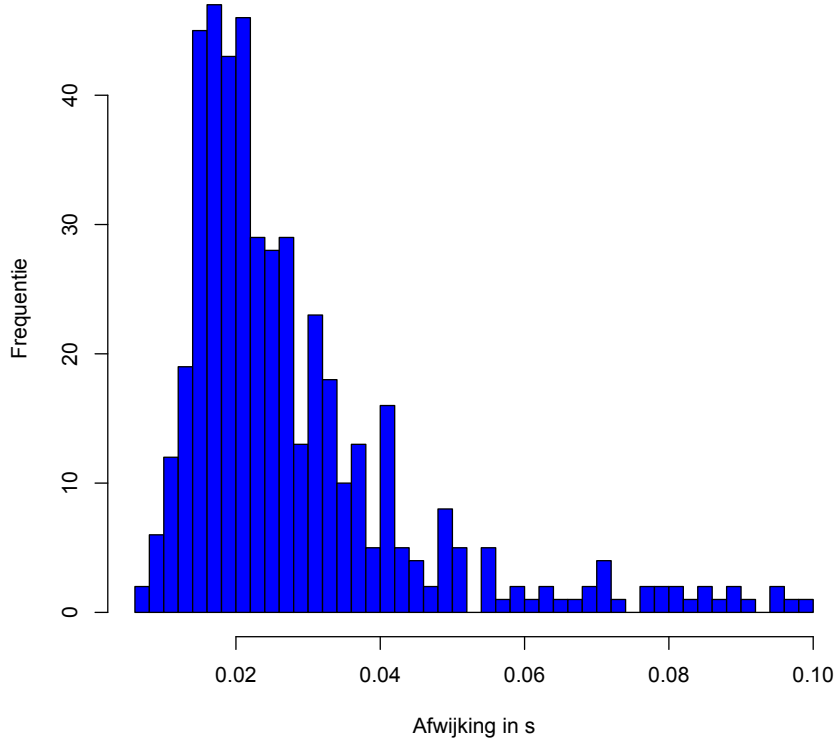
In TIMIT is spraak van personen uit acht verschillende dialectgroepen opgenomen. In elke dialectgroep is zowel mannelijke als vrouwelijke spraak vertegenwoordigd. In deze sectie wordt onderzocht of er tussen de genders en de dialectgroepen een significant verschil bestaat in kwaliteit van automatische annotatie.

Voor de berekening van de verschillen tussen dialectgroepen en de genders wordt wederom de mediaan van de absolute afwijkingen van de grenzen van de automatische annotatie ten opzichte van de handmatige annotatie gebruikt, om afhankelijkheid te voorkomen. Deze zijn per dialectgroep ook niet symmetrisch verdeeld. In figuur 4.2 staat een histogram van de absolute afwijkingen binnen dialectgroep dr1 uit TIMIT van de bestanden geannoteerd met de aligned methode.

Om te testen of de data normaal verdeeld in de groepen binnen de factoren dialect en gender is wederom de Cramer-von Mises test gebruikt. Hieruit bleek dat geen van de groepen binnen de factoren dialect en gender normaal verdeeld is ($p<0{,}001$ voor elke groep).

Omdat de data binnen de groepen niet normaal verdeeld is, is de Kruskal-Wallis test gebruikt. Dit is een uitbreiding van de Mann-Whitney U test, waarbij meer dan twee groepen met elkaar kunnen worden vergeleken, zonder dat de type 1 error toeneemt. De nulhypothese van deze test is dat groepen een gelijke mediaan hebben. De test is afzonderlijk uitgevoerd op elke annotatiemethode (aligned, aligned_corr en aligned_phon). De medianen van de genders bleken niet significant af te wijken in elke methode ($p>0.05$). De medianen van de dialectgroepen in de aligned_corr en aligned_phon methode bleken ook niet significant van elkaar af te wijken ($p>0.05$). De medianen van de dialectgroepen in de aligned methode weken wel significant van elkaar af ($p<0.001$).

Om het multiple comparisons probleem te ontwijken, is een Bonferroni correctie toegepast. In totaal zijn 28 vergelijkingen uitgevoerd, wat betekent dat de maximale p-waarde die nog significant is, moet worden gedeeld door 28. Hierdoor komt voor een p-waarde van 0.05 de maximale p-waarde die significant is uit op $0.05/28 = 0.00179$. Alle resultaten met een p-waarde <0.00179 zijn significant, alle andere waarden niet. De automatische anno-



Figuur 4.2: Een histogram van de verdeling van de afwijkingen van dialectgroep dr1 uit TIMIT, automatisch geannoteerd met de aligned methode. De afwijkingen die zijn gebruikt voor het maken van deze histogram zijn medianen van de afwijkingen van elke opgenomen uiting.

tatie van dialectgroep dr5, bestaande uit sprekers uit de zuidelijke Verenigde Staten, wijkt significant sterker af van de handmatige annotatie dan de annotatie van de dialectgroepen dr2, bestaande uit sprekers uit de noordelijke Verenigde Staten, dr3 bestaande uit sprekers uit de centraal-noordelijke Verenigde Staten en dr7 bestaande uit sprekers uit de westelijke Verenigde Staten. De mediaan van de afwijkingen is echter maar 2 ms groter. Dit verschil is weliswaar significant, maar niet niet heel groot.

Tabel 4.4: De p-waarden voor de Mann-Whitney U test voor de verschillende dialectregio's in de TIMIT aligned conditie. De gekleurde cellen geven aan p-waarden$<$0.00179 aan. Dit geeft aan dat de medianen van twee groepen verschillen. Een p-waarde van 0.000 moet worden geïnterpreteerd als p$<$0.001

	dr2	dr3	dr4	dr5	dr6	dr7	dr8
dr1	0.434	0.322	0.284	0.038	0.711	0.280	0.332
dr2		0.791	0.030	0.001	0.243	0.690	0.698
dr3			0.013	0.000	0.168	0.871	0.834
dr4				0.228	0.527	0.009	0.047
dr5					0.109	0.000	0.004
dr6						0.134	0.191
dr7							0.901

De mediaan en median absolute deviation (MAD, de mediaan van absolute afstanden in een dataset tot de mediaan van de dataset) van de medianen van de absolute afwijkingen van de automatische annotatie ten opzichte van de handmatige annotatie van TIMIT per

Tabel 4.5: De mediaan en MAD voor de afwijkingen van de automatische annotatie tov de handmatige annotatie van het TIMIT corpus, voor zowel de aligned als de aligned_corr methode. De methodes zijn aangegeven met -a en -a_c en -a_p voor aligned aligned_corr en aligned_phn respectievelijk. Alle data in secondes.

dialect	mediaan-a	MAD-a	median-a_c	MAD-a_c	mediaan-a_p	MAD-a_p
dr1	0.023	0.011	0.028	0.016	0.028	0.016
dr2	0.023	0.011	0.028	0.016	0.028	0.017
dr3	0.023	0.011	0.027	0.015	0.027	0.017
dr4	0.024	0.012	0.028	0.015	0.028	0.016
dr5	0.025	0.013	0.028	0.015	0.027	0.015
dr6	0.024	0.011	0.029	0.016	0.029	0.018
dr7	0.023	0.010	0.027	0.015	0.028	0.017
dr8	0.024	0.012	0.028	0.017	0.026	0.015

bestand staan in tabel 4.4

4.3.2 Het IFA corpus

Tussen de mannelijke en vrouwelijke sprekers in het IFA corpus bestaat geen significant verschil in afwijking van de automatische annotatie ten opzichte van de handmatige ($p>0{,}05$). De sprekers in IFA zijn niet opgedeeld in verschillende dialectgroepen, en van de 8 sprekers is niet in het corpus opgenomen tot welk Nederlands dialect deze behoren. Hierdoor is geen vergelijking tussen dialectgroepen mogelijk. Wel kan worden bekeken of het verschil in afwijking van de automatische annotatie tussen sprekers significant is. In tabel 4.6 zijn de medianen en MADs elke spreker in het IFA opgenomen voor beide methodes.

De Kruskal-Wallis test op deze data levert voor zowel de aligned als de aligned_corr methode een significant resultaat op. Voor beide methodes is de data van elk paar sprekers met elkaar vergeleken op dezelfde wijze als in paragraaf 4.3.1. Het significantieniveau is ook hier weer aangepast naar 0,00179. Hieruit blijkt dat de automatische annotatie voor spreker O significant sterker afwijkt van de handmatige annotatie dan de andere sprekers. Bovendien is de annotatie van spreker R significant beter dan die van andere sprekers in de alligned_corr methode, en significant beter dan sprekers G, K, en O in de aligned methode.

4.4 Verschillen annotatie

In deze sectie wordt besproken hoe sterk de automatische annotatie afwijkt van de handmatige annotaties.

Tabel 4.6: De mediaan en MAD voor de afwijkingen van de automatische annotatie tov de handmatige annotatie van het IFA corpus, voor zowel de aligned als de aligned_corr methode. De methodes zijn aangegeven met -a en -a_c voor aligned en aligned_corr respectievelijk. Alle data is seconden.

spreker	mediaan-a	MAD-a	mediaan-a_c	MAD-a_c
E	0.025	0.026	0.033	0.035
G	0.027	0.029	0.034	0.036
K	0.028	0.028	0.032	0.036
L	0.025	0.026	0.031	0.031
H	0.024	0.026	0.033	0.035
N	0.026	0.026	0.031	0.034
O	0.037	0.038	0.037	0.039
R	0.024	0.024	0.027	0.027

4.4.1 Matchende foneemgrenzen

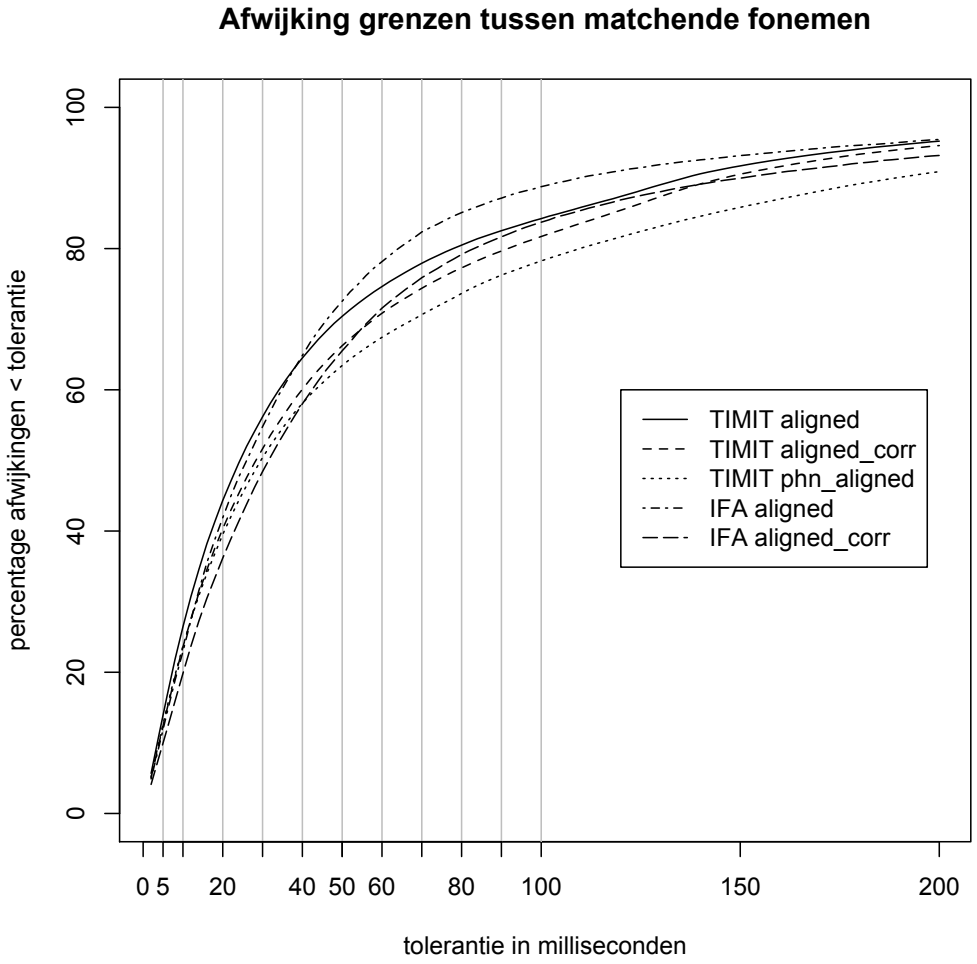
Om de hoofdvraag van deze scriptie te kunnen beantwoorden, is het nodig om te weten hoeveel handmatige correctie nodig is na de automatische annotatie. Als maat hiervoor wordt gekeken hoeveel procent van de grenzen tussen segmenten onder een bepaalde afwijking (toleranties) vallen. In deze sectie wordt eerst alleen naar grenzen tussen matchende segmenten gekeken. Inserties en deleties worden in de volgende sectie behandeld. In tegenstelling tot in de vorige sectie worden alle foneemgrenzen gebruikt, in plaats van de mediaan van de afwijkingen per bestand. De reden hiervoor is dat dit een betere indicatie geeft van hoeveel werk nodig is in een mogelijke nacontrole.

In figuur 4.3 is een grafiek te zien waarin de resultaten voor de twee corpora met de verschillende methodes zijn opgenomen. In deze grafiek is te zien dat in zowel het TIMIT corpus als het IFA corpus de aligned methode resultaten oplevert die minder afwijken van een handmatige annotatie.

In tabellen 4.7 en 4.8 zijn deze resultaten opgesplitst naar foneemovergang. In de tabellen is te zien dat de afwijkingen per overgang sterk uiteenlopen. Annotatie van foneemovergangen van en naar stiltes wijken in beide corpora in alle methoden sterk af van de handmatige annotatie. Dit zijn echter weinig frequente foneemovergangen, waardoor het effect op de totale annotatie beperkt blijft.

De foneemovergangen die in alle corpora vaak voorkomen – alle overgangen van en naar klinkers, behalve V>S en S>V – scoren in alle corpora met alle methodes in het eerste en tweede kwartiel. Opvallend is dat de annotatie van een N>F overgang in het TIMIT corpus minder sterk afwijkt van de handmatige annotatie dan in het IFA corpus.

Oorzaken van de verschillen in annotatie worden in de volgende sectie besproken.



Figuur 4.3: Het percentage afwijkingen van grenzen tussen matchende fonemen ten opzichte van de handmatige annotatie dat onder een bepaalde tolerantie valt.

Tabel 4.7: Het percentage afwijkingen voor een grens tussen een overgang tussen twee foneemklassen. A = approximant, F = fricatief, N = nasaal, P = plosief, V = klinker, S = stilte. De kleuren van de cellen geven aan welk percentage van de grenzen op een overgang (rijen) een afwijking ten opzichte van het handmatig geannoteerde corpus hebben, die kleiner is dan de tolerantie in de kolomhoofden. Rood betekent < 33%, Geel betekent 33-66%, Groen betekent >66%. Des te verder het groen naar links uitstrekt, des te beter de uitlijning voor die foneemovergang de handmatige uitlijning benadert. De % tot kolom bevat het percentage per transities van alle transities.

TIMIT: Woordelijke input, zonder correctie spreeksnelheid

Cumulatieve percentages boundaryafwijkingen < tolerantie in ms

Transitie	% tot	<0.005	<0.010	<0.020	<0.030	<0.040	<0.050	<0.10	<0.50	>=0.5	MT
P>N	0.15%	26.5%	50.6%	72.3%	82.6%	87.0%	90.5%	96.4%	100.0%	0.0%	75.7%
F>S	0.31%	24.6%	42.6%	67.0%	76.4%	81.8%	84.3%	90.0%	100.0%	0.0%	70.8%
S>N	0.53%	27.8%	45.0%	61.2%	70.5%	75.5%	79.6%	89.1%	99.9%	0.1%	68.6%
A>A	0.44%	13.6%	28.3%	51.1%	67.2%	77.3%	84.0%	96.4%	99.9%	0.1%	64.7%
V>F	9.23%	23.1%	38.5%	56.6%	65.6%	71.2%	74.9%	85.1%	99.6%	0.4%	64.3%
N>F	1.44%	17.9%	32.6%	53.8%	65.2%	72.7%	77.1%	86.7%	99.7%	0.3%	63.2%
F>V	9.64%	18.6%	35.8%	55.0%	63.9%	69.2%	72.8%	85.6%	99.6%	0.4%	62.6%
F>N	0.41%	14.6%	27.0%	49.9%	64.0%	74.3%	79.1%	89.0%	99.7%	0.3%	62.2%
A>F	0.29%	14.4%	31.4%	53.4%	65.7%	72.5%	76.6%	81.5%	99.8%	0.2%	61.9%
P>V	14.15%	16.2%	32.6%	53.3%	63.5%	69.3%	73.1%	82.9%	99.5%	0.5%	61.3%
V>P	12.09%	16.7%	30.7%	50.2%	62.5%	69.3%	73.7%	84.9%	99.6%	0.4%	61.0%
N>N	0.13%	15.4%	27.1%	44.3%	62.0%	69.7%	75.6%	92.8%	100.0%	0.0%	60.9%
F>F	0.77%	12.0%	24.7%	46.0%	62.0%	71.5%	78.0%	90.8%	99.5%	0.5%	60.6%
S>A	0.57%	16.7%	29.6%	47.6%	59.0%	66.9%	71.8%	83.6%	99.5%	0.5%	59.3%
P>A	2.38%	11.9%	24.4%	44.5%	58.8%	69.1%	76.7%	88.3%	99.6%	0.4%	59.2%
F>A	0.76%	13.6%	26.4%	45.4%	58.7%	69.0%	74.7%	85.4%	99.5%	0.5%	59.1%
A>P	0.49%	11.6%	24.3%	42.8%	59.0%	68.0%	74.7%	88.7%	99.5%	0.5%	58.6%
P>F	1.39%	11.4%	21.5%	40.8%	57.9%	68.9%	76.5%	90.5%	99.7%	0.3%	58.4%
N>P	2.24%	14.5%	26.6%	43.2%	54.5%	63.1%	71.3%	88.0%	99.7%	0.3%	57.6%
N>A	0.43%	12.4%	22.5%	40.5%	56.2%	65.3%	71.3%	86.5%	99.9%	0.1%	56.8%
P>P	0.39%	14.4%	24.8%	40.9%	52.6%	61.4%	70.1%	90.3%	100.0%	0.0%	56.8%
V>N	8.75%	13.0%	24.3%	42.3%	54.5%	63.5%	69.7%	84.8%	99.6%	0.4%	56.5%
A>V	7.01%	9.7%	19.6%	37.5%	52.8%	65.0%	73.5%	88.6%	99.7%	0.3%	55.8%
V>V	6.09%	9.6%	19.0%	36.1%	50.3%	61.2%	68.8%	84.5%	99.6%	0.4%	53.6%
S>V	1.54%	9.7%	21.4%	37.0%	49.6%	60.5%	66.6%	83.4%	99.4%	0.6%	53.4%
N>V	4.16%	5.7%	12.6%	27.9%	45.0%	58.5%	68.3%	88.2%	99.7%	0.3%	50.7%
A>N	0.06%	2.0%	7.9%	28.7%	47.5%	60.4%	66.3%	92.1%	100.0%	0.0%	50.6%
F>P	2.47%	7.1%	15.1%	28.7%	42.3%	55.2%	65.8%	88.1%	99.5%	0.5%	50.2%
V>A	3.69%	8.8%	16.5%	31.1%	42.7%	52.7%	61.9%	86.0%	99.7%	0.3%	49.9%
N>S	0.37%	7.2%	13.2%	27.4%	40.0%	54.7%	66.6%	85.2%	100.0%	0.0%	49.3%
V>S	3.57%	6.8%	13.7%	26.9%	39.6%	52.0%	61.9%	79.2%	99.7%	0.3%	47.5%
P>S	0.03%	7.0%	18.6%	30.2%	39.5%	44.2%	51.2%	69.8%	100.0%	0.0%	45.1%
A>S	0.15%	4.0%	11.6%	20.8%	34.4%	41.6%	45.6%	56.8%	99.6%	0.4%	39.3%
S>F	2.46%	5.5%	11.2%	23.1%	32.0%	39.8%	44.9%	56.0%	99.6%	0.4%	39.0%
S>P	1.08%	6.8%	12.7%	18.4%	22.4%	25.6%	27.4%	53.3%	99.1%	0.9%	33.2%
S>S	0.35%	2.7%	3.9%	7.3%	9.9%	14.5%	21.7%	57.0%	99.5%	0.5%	27.1%
Totaal		14.0%	26.4%	44.3%	56.2%	64.5%	70.4%	84.2%	99.6%	0.0%	57.5%

Totaal aantal transities: 168270

TIMIT: Woordelijke input, met correctie spreeksnelheid

Cumulatieve percentages boundaryafwijkingen < tolerantie in ms

Transitie	% tot	<0.005	<0.010	<0.020	<0.030	<0.040	<0.050	<0.10	<0.50	>=0.5	MT
P>N	0.15%	20.8%	35.2%	53.6%	66.0%	74.8%	79.2%	90.4%	100.0%	0.0%	65.0%
N>F	1.45%	17.4%	32.4%	53.8%	65.7%	72.0%	76.6%	85.8%	99.5%	0.5%	62.9%
F>N	0.41%	12.8%	27.7%	47.9%	63.2%	73.1%	79.0%	90.5%	99.9%	0.1%	61.8%
V>F	9.29%	19.3%	33.9%	52.0%	61.8%	68.3%	72.4%	83.6%	99.6%	0.4%	61.4%
A>F	0.29%	16.4%	30.9%	54.2%	64.0%	70.6%	72.8%	81.2%	100.0%	0.0%	61.2%
F>V	9.71%	18.0%	32.8%	51.2%	59.9%	65.8%	69.7%	83.7%	99.6%	0.4%	60.1%
F>F	0.77%	12.0%	23.3%	44.7%	60.4%	70.2%	77.3%	89.7%	99.3%	0.7%	59.6%
N>N	0.13%	14.2%	26.5%	46.6%	58.9%	68.5%	72.6%	89.5%	99.5%	0.5%	59.5%
S>N	0.52%	16.8%	31.0%	48.6%	57.6%	64.4%	71.0%	86.7%	99.5%	0.5%	59.4%
F>S	0.31%	13.9%	26.8%	47.9%	59.5%	66.6%	72.8%	85.5%	99.8%	0.2%	59.1%
A>A	0.45%	12.5%	23.9%	43.0%	55.9%	65.7%	75.1%	93.5%	99.9%	0.1%	58.7%
P>V	14.19%	14.6%	29.6%	48.1%	58.0%	64.7%	69.0%	80.5%	99.6%	0.4%	58.0%
F>A	0.76%	13.2%	24.7%	44.3%	56.9%	65.7%	72.5%	85.9%	99.5%	0.5%	57.8%
A>P	0.49%	11.4%	24.6%	44.4%	56.8%	64.4%	70.1%	86.7%	99.8%	0.2%	57.3%
V>P	12.07%	13.8%	25.6%	43.6%	55.5%	63.3%	68.3%	81.5%	99.6%	0.4%	56.4%
P>F	1.40%	10.7%	20.0%	36.9%	52.8%	64.0%	72.3%	88.0%	99.9%	0.1%	55.6%
N>P	2.26%	13.6%	24.9%	40.4%	50.6%	59.7%	66.8%	86.1%	99.6%	0.4%	55.2%
N>A	0.44%	10.7%	21.3%	38.0%	51.9%	60.4%	69.0%	84.4%	99.6%	0.4%	54.4%
P>P	0.39%	12.1%	20.8%	39.4%	50.3%	58.3%	65.5%	88.8%	99.8%	0.2%	54.4%
V>N	8.76%	11.7%	22.2%	39.0%	51.0%	60.1%	66.7%	82.7%	99.6%	0.4%	54.1%
A>N	0.06%	4.0%	11.1%	31.3%	53.5%	63.6%	70.7%	92.9%	100.0%	0.0%	53.4%
S>A	0.56%	15.5%	23.6%	39.5%	49.6%	56.7%	62.7%	79.6%	99.7%	0.3%	53.4%
P>A	2.38%	9.8%	20.7%	37.2%	48.3%	57.4%	66.5%	84.8%	99.6%	0.4%	53.1%
A>V	7.00%	8.9%	17.6%	33.9%	48.2%	59.1%	67.8%	85.9%	99.6%	0.4%	52.6%
S>V	1.56%	8.8%	18.3%	33.3%	45.8%	56.0%	63.2%	82.0%	99.5%	0.5%	50.9%
V>V	6.03%	7.8%	15.8%	31.1%	44.0%	54.2%	62.0%	79.4%	99.6%	0.4%	49.2%
F>P	2.49%	7.4%	14.2%	27.6%	39.7%	51.9%	62.3%	87.5%	99.7%	0.3%	48.8%
N>V	4.13%	5.6%	11.4%	25.8%	40.8%	53.9%	63.5%	85.0%	99.6%	0.4%	48.2%
N>S	0.37%	6.5%	13.1%	26.1%	40.2%	52.4%	61.1%	81.0%	99.8%	0.2%	47.5%
V>A	3.69%	6.8%	13.6%	27.6%	38.8%	48.7%	57.8%	83.2%	99.7%	0.3%	47.0%
V>S	3.48%	7.4%	13.4%	26.7%	39.1%	50.0%	58.5%	76.1%	99.6%	0.4%	46.4%
P>S	0.03%	16.3%	20.4%	32.7%	38.8%	46.9%	49.0%	65.3%	100.0%	0.0%	46.2%
A>S	0.15%	6.0%	12.0%	19.3%	29.7%	43.0%	47.0%	55.8%	99.6%	0.4%	39.1%
S>F	2.45%	5.3%	10.7%	22.1%	30.9%	37.4%	42.2%	54.8%	99.6%	0.4%	37.9%
S>P	1.08%	7.7%	13.5%	19.2%	22.3%	25.1%	27.3%	44.8%	99.1%	0.9%	32.3%
S>S	0.27%	1.7%	3.0%	7.0%	10.7%	16.1%	24.6%	57.6%	100.0%	0.0%	27.6%
Totaal		12.4%	23.6%	40.3%	51.6%	60.1%	66.2%	81.7%	99.6%	0.0%	54.5%

Totaal aantal transities: 167302

Tabel 4.7: vervolg

TIMIT: Foneemstring input

Cumulatieve percentages boundaryafwijkingen < tolerantie in ms

Transitie	% tot	<0.005	<0.010	<0.020	<0.030	<0.040	<0.050	<0.10	<0.50	>=0.5	MT
A>F	0.17%	24.7%	40.6%	56.3%	62.8%	67.2%	69.7%	79.1%	99.4%	0.6%	62.5%
V>F	8.61%	23.0%	37.8%	53.5%	62.4%	67.9%	71.7%	81.4%	99.3%	0.7%	62.1%
P>N	0.42%	20.8%	37.1%	53.5%	62.4%	67.3%	70.4%	80.2%	99.4%	0.6%	61.4%
F>N	0.60%	14.0%	27.6%	49.7%	60.7%	69.2%	74.0%	83.8%	99.6%	0.4%	59.8%
F>V	9.20%	17.8%	33.0%	50.1%	59.4%	65.6%	69.6%	80.0%	99.3%	0.7%	59.4%
N>F	1.46%	18.8%	31.7%	47.7%	57.5%	63.5%	67.9%	78.1%	99.4%	0.6%	58.1%
P>F	1.37%	12.9%	24.1%	41.6%	56.0%	65.0%	70.8%	81.7%	99.5%	0.5%	56.4%
P>V	13.85%	14.2%	27.7%	44.6%	54.2%	60.6%	65.2%	78.8%	99.4%	0.6%	55.6%
V>V	4.88%	11.0%	21.4%	40.3%	54.4%	63.8%	69.2%	81.6%	99.5%	0.5%	55.1%
N>N	0.16%	10.9%	22.5%	41.3%	53.9%	61.8%	67.2%	81.6%	99.7%	0.3%	54.9%
F>A	0.75%	13.4%	25.9%	41.4%	51.6%	59.4%	66.2%	79.9%	99.0%	1.0%	54.6%
V>P	11.99%	13.2%	24.6%	41.2%	52.8%	60.3%	65.0%	77.2%	99.4%	0.6%	54.2%
A>P	0.47%	11.1%	20.0%	39.9%	52.6%	61.2%	66.1%	81.9%	99.1%	0.9%	54.0%
A>A	0.09%	12.5%	21.3%	38.1%	50.0%	59.4%	69.4%	81.9%	99.4%	0.6%	54.0%
V>A	2.82%	11.0%	20.3%	38.3%	50.4%	59.7%	67.8%	84.1%	99.5%	0.5%	53.9%
V>N	8.07%	11.9%	22.9%	39.2%	51.4%	59.8%	65.9%	80.0%	99.4%	0.6%	53.8%
N>P	2.29%	9.9%	19.1%	34.7%	47.4%	59.0%	66.6%	81.6%	99.4%	0.6%	52.2%
A>V	6.65%	9.3%	18.6%	35.7%	48.2%	57.5%	63.9%	79.2%	99.5%	0.5%	51.5%
N>V	4.52%	8.1%	15.7%	31.7%	47.2%	58.8%	67.1%	82.5%	99.4%	0.6%	51.3%
F>F	0.75%	9.7%	18.1%	34.4%	48.0%	56.9%	64.6%	79.1%	98.9%	1.1%	51.2%
N>A	0.47%	10.6%	19.1%	35.2%	48.1%	56.4%	63.3%	77.0%	99.3%	0.7%	51.1%
V>S	3.77%	10.9%	21.0%	36.0%	46.4%	53.1%	58.5%	73.6%	99.6%	0.4%	49.9%
P>A	2.21%	8.2%	17.1%	33.6%	46.6%	56.2%	62.1%	74.3%	99.3%	0.7%	49.7%
P>P	0.38%	10.0%	20.4%	34.0%	43.0%	51.6%	57.1%	77.7%	99.3%	0.7%	49.1%
S>N	0.81%	9.9%	19.2%	32.5%	39.8%	46.3%	51.0%	78.8%	99.3%	0.7%	47.1%
F>P	2.37%	7.9%	15.5%	29.0%	39.8%	49.0%	57.2%	78.2%	99.5%	0.5%	47.0%
F>S	0.93%	7.5%	15.8%	29.8%	41.6%	49.4%	53.9%	63.2%	99.4%	0.6%	45.1%
S>V	2.89%	7.1%	14.0%	28.1%	38.8%	46.4%	52.2%	71.2%	99.4%	0.6%	44.6%
P>S	0.38%	8.1%	15.3%	28.8%	36.9%	45.8%	52.3%	69.5%	99.4%	0.6%	44.5%
N>S	1.01%	7.4%	14.5%	26.4%	35.5%	44.2%	50.3%	71.8%	99.7%	0.3%	43.7%
A>N	0.03%	3.1%	7.7%	18.5%	26.2%	36.9%	53.8%	78.5%	100.0%	0.0%	40.6%
S>A	0.80%	4.4%	8.5%	18.4%	26.8%	34.2%	39.2%	70.6%	99.1%	0.9%	37.6%
A>S	0.12%	5.3%	12.8%	20.8%	29.6%	37.2%	38.1%	49.1%	99.6%	0.4%	36.6%
S>S	0.68%	7.6%	11.8%	16.5%	22.7%	29.9%	37.4%	64.7%	99.4%	0.6%	36.2%
S>F	2.47%	5.2%	10.2%	17.4%	23.8%	29.6%	34.5%	68.6%	99.3%	0.7%	36.1%
S>P	1.57%	2.9%	4.8%	7.8%	13.0%	17.1%	18.9%	66.7%	99.0%	1.0%	28.8%
Totaal		12.7%	23.7%	39.5%	50.4%	58.1%	63.4%	78.2%	99.4%	0.0%	53.2%
								Totaal aantal transities:			187542

Tabel 4.8: Het percentage afwijkingen voor een grens tussen een overgang tussen twee foneemklassen. A = approximant, F = fricatief, N = nasaal, P = plosief, V = klinker, S = stilte. De kleuren van de cellen geven aan welk percentage van de grenzen op een overgang (rijen) een afwijking ten opzichte van het handmatig geannoteerde corpus hebben, die kleiner is dan de tolerantie in de kolomhoofden. Rood betekent < 33%, Geel betekent 33-66%, Groen betekent >66%. Des te verder het groen naar links uitstrekt, des te beter de uitlijning voor die foneemovergang de handmatige uitlijning benadert. De % tot kolom bevat het percentage per transities van alle transities.

TIMIT: Woordelijke input, zonder correctie spreeksnelheid

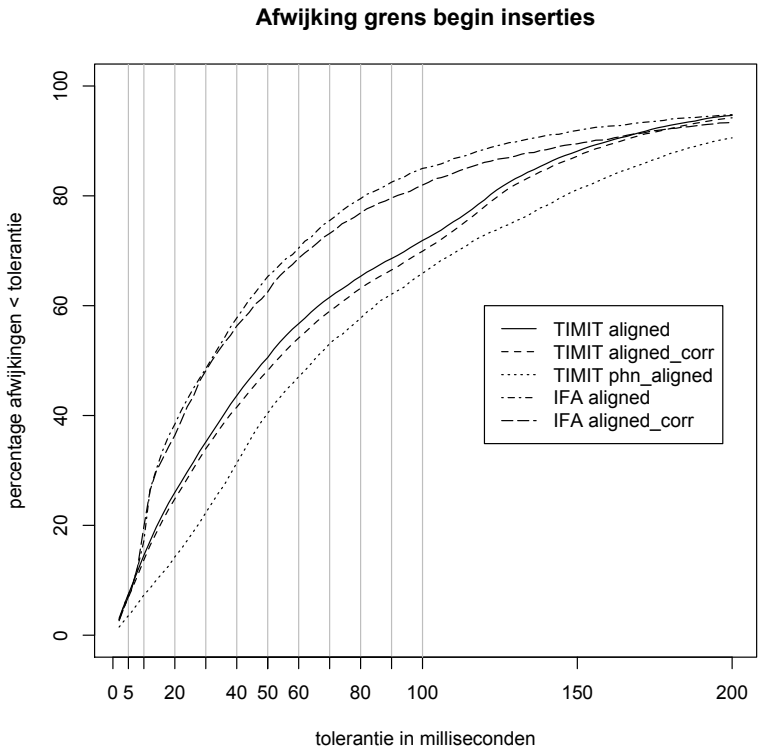
Cumulatieve percentages boundaryafwijkingen < tolerantie in ms

Transitie	% tot	<0.005	<0.01	<0.02	<0.03	<0.04	<0.05	<0.1	<0.5	>=0.5	MT
S>A	0.1%	0.0%	0.0%	97.3%	97.3%	100.0%	100.0%	100.0%	100.0%	0.0%	74.3%
S>F	1.4%	1.0%	2.2%	87.1%	88.5%	89.5%	90.9%	96.7%	99.5%	0.5%	69.4%
F>N	0.2%	28.6%	34.7%	49.0%	69.4%	79.6%	87.8%	93.9%	100.0%	0.0%	67.9%
P>N	0.4%	16.5%	33.9%	58.7%	70.2%	78.5%	81.0%	89.3%	97.5%	2.5%	65.7%
N>A	0.6%	17.6%	30.9%	52.1%	67.6%	76.1%	83.5%	89.4%	100.0%	0.0%	64.6%
V>F	8.4%	17.2%	32.3%	54.5%	67.9%	75.3%	80.0%	89.7%	99.4%	0.6%	64.5%
F>A	1.1%	17.6%	33.0%	54.9%	66.0%	73.5%	81.5%	90.1%	98.1%	1.9%	64.4%
F>V	10.0%	17.3%	32.8%	54.6%	66.6%	74.1%	79.7%	89.8%	99.4%	0.6%	64.3%
P>V	13.1%	18.4%	34.2%	55.5%	66.1%	72.9%	78.0%	88.7%	99.2%	0.8%	64.1%
N>N	0.3%	13.7%	31.6%	51.6%	61.1%	74.7%	81.1%	93.7%	100.0%	0.0%	63.4%
A>A	0.4%	15.5%	24.5%	48.2%	60.9%	75.5%	82.7%	92.7%	100.0%	0.0%	62.5%
N>F	1.7%	14.5%	27.6%	50.5%	64.4%	72.6%	77.9%	87.8%	98.9%	1.1%	61.8%
V>N	11.9%	14.2%	27.5%	48.7%	62.8%	71.8%	77.5%	88.5%	99.1%	0.9%	61.3%
P>A	2.4%	15.8%	29.5%	49.7%	59.8%	70.0%	74.8%	88.7%	98.8%	1.2%	60.9%
F>F	0.7%	10.2%	24.4%	42.2%	59.6%	71.6%	79.1%	93.3%	99.1%	0.9%	59.9%
F>S	0.1%	16.7%	19.4%	36.1%	66.7%	72.2%	72.2%	86.1%	97.2%	2.8%	58.3%
A>F	1.6%	11.6%	20.8%	43.2%	57.3%	68.4%	76.0%	89.3%	99.6%	0.4%	58.3%
S>V	0.6%	12.4%	21.5%	35.6%	62.7%	70.6%	76.3%	85.3%	97.7%	2.3%	57.8%
S>N	0.1%	7.9%	26.3%	52.6%	57.9%	60.5%	73.7%	78.9%	94.7%	5.3%	56.6%
F>P	2.5%	8.7%	17.3%	33.8%	50.9%	65.6%	77.5%	91.3%	99.2%	0.8%	55.5%
V>A	10.1%	9.8%	19.8%	36.5%	50.5%	62.1%	69.8%	88.3%	99.6%	0.4%	54.6%
N>V	5.2%	8.4%	17.6%	35.0%	49.9%	62.8%	72.8%	88.4%	98.8%	1.2%	54.2%
P>F	1.3%	7.1%	14.5%	31.8%	48.3%	62.1%	72.7%	92.6%	99.0%	1.0%	53.5%
A>N	0.2%	4.9%	19.7%	32.8%	47.5%	57.4%	70.5%	91.8%	100.0%	0.0%	53.1%
V>S	0.1%	9.5%	19.0%	33.3%	40.5%	52.4%	66.7%	90.5%	100.0%	0.0%	51.5%
A>V	9.8%	7.1%	15.0%	28.9%	44.6%	58.5%	69.1%	88.9%	99.2%	0.8%	51.4%
V>V	0.1%	4.9%	14.6%	31.7%	43.9%	63.4%	70.7%	82.9%	97.6%	2.4%	51.2%
N>S	0.1%	15.8%	21.1%	26.3%	34.2%	44.7%	63.2%	86.8%	97.4%	2.6%	48.7%
A>S	0.1%	3.0%	12.1%	27.3%	48.5%	54.5%	60.6%	72.7%	90.9%	9.1%	46.2%
V>P	7.8%	4.8%	10.3%	22.0%	33.8%	45.3%	56.4%	88.1%	98.9%	1.1%	45.0%
P>S	0.1%	7.3%	12.2%	22.0%	39.0%	46.3%	58.5%	70.7%	97.6%	2.4%	44.2%
N>P	3.6%	2.9%	7.1%	17.0%	28.1%	44.0%	58.8%	86.3%	99.1%	0.9%	42.9%
A>P	2.2%	2.5%	5.9%	12.6%	21.3%	31.6%	43.2%	81.2%	99.7%	0.3%	37.2%
P>P	1.2%	1.9%	4.0%	10.9%	18.9%	29.6%	43.2%	87.5%	100.0%	0.0%	37.0%
S>P	0.5%	1.9%	4.4%	8.8%	11.9%	33.3%	39.6%	88.7%	98.1%	1.9%	35.8%
Totaal		11.8%	22.9%	41.9%	54.8%	64.9%	72.5%	88.8%	99.2%	0.8%	57.1%
									Totaal aantal transities:		30430

TIMIT: Woordelijke input, met correctie spreeksnelheid

Cumulatieve percentages boundaryafwijkingen < tolerantie in ms

Transitie	% tot	<0.005	<0.01	<0.02	<0.03	<0.04	<0.05	<0.1	<0.5	>=0.5	MT
S>A	0.1%	0.0%	0.0%	97.3%	97.3%	100.0%	100.0%	100.0%	100.0%	0.0%	74.3%
S>F	1.4%	1.0%	2.2%	87.1%	88.5%	89.5%	90.9%	96.7%	99.5%	0.5%	69.4%
N>N	0.3%	28.6%	34.7%	49.0%	69.4%	79.6%	87.8%	93.9%	100.0%	0.0%	67.9%
P>A	2.4%	16.5%	33.9%	58.7%	70.2%	78.5%	81.0%	89.3%	97.5%	2.5%	65.7%
F>N	0.2%	17.6%	30.9%	52.1%	67.6%	76.1%	83.5%	89.4%	100.0%	0.0%	64.6%
N>A	0.6%	17.2%	32.3%	54.5%	67.9%	75.3%	80.0%	89.7%	99.4%	0.6%	64.5%
F>V	10.0%	17.6%	33.0%	54.9%	66.0%	73.5%	81.5%	90.1%	98.1%	1.9%	64.4%
P>N	0.4%	17.3%	32.8%	54.6%	66.6%	74.1%	79.7%	89.8%	99.4%	0.6%	64.3%
F>F	0.8%	18.4%	34.2%	55.5%	66.1%	72.9%	78.0%	88.7%	99.2%	0.8%	64.1%
V>F	8.3%	13.7%	31.6%	51.6%	61.1%	74.7%	81.1%	93.7%	100.0%	0.0%	63.4%
P>V	13.0%	15.5%	24.5%	48.2%	60.9%	75.5%	82.7%	92.7%	100.0%	0.0%	62.5%
N>F	1.7%	14.5%	27.6%	50.5%	64.4%	72.6%	77.9%	87.8%	98.9%	1.1%	61.8%
F>S	0.1%	14.2%	27.5%	48.7%	62.8%	71.8%	77.5%	88.5%	99.1%	0.9%	61.3%
F>A	1.1%	15.8%	29.5%	49.7%	59.8%	70.0%	74.8%	88.7%	98.8%	1.2%	60.9%
V>N	11.9%	10.2%	24.4%	42.2%	59.6%	71.6%	79.1%	93.3%	99.1%	0.9%	59.9%
A>A	0.4%	16.7%	19.4%	36.1%	66.7%	72.2%	72.2%	86.1%	97.2%	2.8%	58.3%
F>P	2.4%	11.6%	20.8%	43.2%	57.3%	68.4%	76.0%	89.3%	99.6%	0.4%	58.3%
S>V	0.6%	12.4%	21.5%	35.6%	62.7%	70.6%	76.3%	85.3%	97.7%	2.3%	57.8%
A>F	1.6%	7.9%	26.3%	52.6%	57.9%	60.5%	73.7%	78.9%	94.7%	5.3%	56.6%
P>F	1.4%	8.7%	17.3%	33.8%	50.9%	65.6%	77.5%	91.3%	99.2%	0.8%	55.5%
V>A	10.0%	9.8%	19.8%	36.5%	50.5%	62.1%	69.8%	88.3%	99.6%	0.4%	54.6%
S>N	0.1%	8.4%	17.6%	35.0%	49.9%	62.8%	72.8%	88.4%	98.8%	1.2%	54.2%
V>S	0.1%	7.1%	14.5%	31.8%	48.3%	62.1%	72.7%	92.6%	99.0%	1.0%	53.5%
A>N	0.2%	4.9%	19.7%	32.8%	47.5%	57.4%	70.5%	91.8%	100.0%	0.0%	53.1%
N>V	5.1%	9.5%	19.0%	33.3%	40.5%	52.4%	66.7%	90.5%	100.0%	0.0%	51.5%
A>V	9.8%	7.1%	15.0%	28.9%	44.6%	58.5%	69.1%	88.9%	99.2%	0.8%	51.4%
V>V	0.1%	4.9%	14.6%	31.7%	43.9%	63.4%	70.7%	82.9%	97.6%	2.4%	51.2%
V>P	7.8%	15.8%	21.1%	26.3%	34.2%	44.7%	63.2%	86.8%	97.4%	2.6%	48.7%
A>S	0.1%	3.0%	12.1%	27.3%	48.5%	54.5%	60.6%	72.7%	90.9%	9.1%	46.2%
N>P	3.6%	4.8%	10.3%	22.0%	33.8%	45.3%	56.4%	88.1%	98.9%	1.1%	45.0%
P>S	0.1%	7.3%	12.2%	22.0%	39.0%	46.3%	58.5%	70.7%	97.6%	2.4%	44.2%
N>S	0.1%	2.9%	7.1%	17.0%	28.1%	44.0%	58.8%	86.3%	99.1%	0.9%	42.9%
A>P	2.2%	2.5%	5.9%	12.6%	21.3%	31.6%	43.2%	81.2%	99.7%	0.3%	37.2%
P>P	1.2%	1.9%	4.0%	10.9%	18.9%	29.6%	43.2%	87.5%	100.0%	0.0%	37.0%
S>P	0.6%	1.9%	4.4%	8.8%	11.9%	33.3%	39.6%	88.7%	98.1%	1.9%	35.8%
Totaal		11.8%	22.9%	41.9%	54.8%	64.9%	72.5%	88.8%	99.2%	0.8%	57.1%
									Totaal aantal transities:		29818



Figuur 4.4: De afwijkingen van de grens aan het begin van een geïnserteerd of een reeks geïnserteerde segmenten ten opzichte van de eindgrens met het segment in de handmatige annotatie dat matcht met het segment direct voor de reeks geïnserteerde segmenten. Deze grafiek geeft het percentage van deze afwijkingen aan dat onder een bepaalde afwijking valt.

4.4.2 Inserties

De methode van het verkrijgen van data over inserties is besproken in het vorige hoofdstuk. De op deze manier verzamelde data is te zien in figuren 4.4 en 4.5. Figuur 4.4 heeft betrekking op de grens links van een insertie, in figuur 3.4 zou dit de afwijking zijn van de grens tussen het [n]-segment en het [t]-segment in de automatische annotatie ten opzichte van de grens tussen het [n]-segment en het [æ]-segment in de handmatige annotatie.

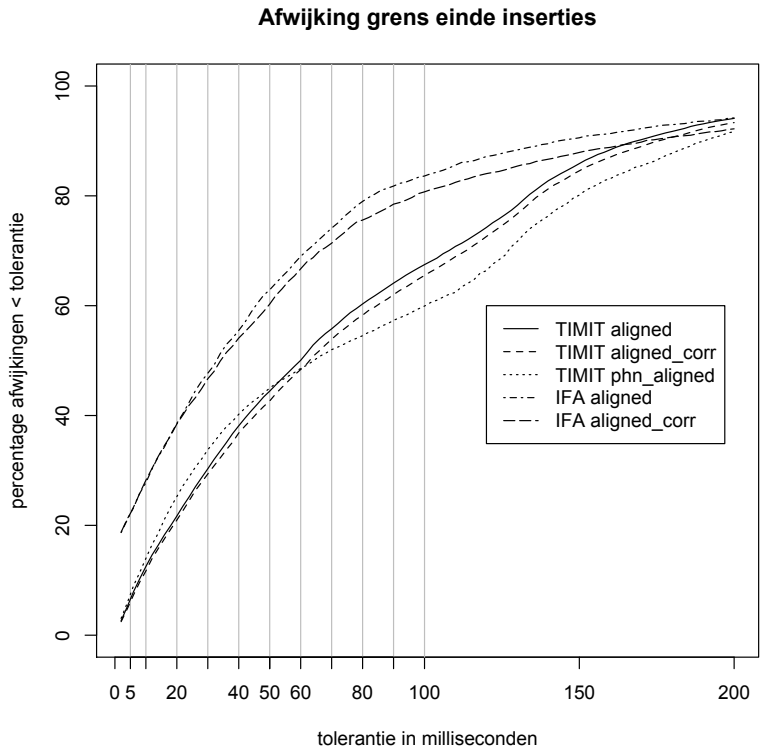
Figuur 4.4 heeft betrekking op de grens rechts van de insertie. In figuur 3.4 zou dit de afwijking zijn van de grens tussen het [t]-segment en het [æ]-segment ten opzichte van de grens tussen het [n]-segment en het [æ]-segment in de handmatige annotatie.

In figuren 4.4 en 4.5 is te zien dat de grenzen links en rechts van inserties sterk afwijken van de handmatige annotatie.

4.5 Oorzaken van afwijkingen

4.5.1 Stiltes en gevulde pauzes

Een belangrijke oorzaak van afwijkingen in de annotaties zijn stiltes in het inputsignaal. Aan het begin en eind van geluidssignalen wordt al een stiltedetectie gebruikt. Deze classificeert geluiden van inademing of uitademing die voor of na een uiting voorkomen als niet-stil, als



Figuur 4.5: De afwijkingen van de grens aan het eind van een geïnserteerd segment of een reeks geïnserteerde segmenten ten opzichte van de eindgrens met het segment in de handmatige annotatie dat matcht met het segment direct na de reeks geïnserteerde segmenten. Deze grafiek geeft het percentage van deze afwijkingen aan dat onder een bepaalde afwijking valt.

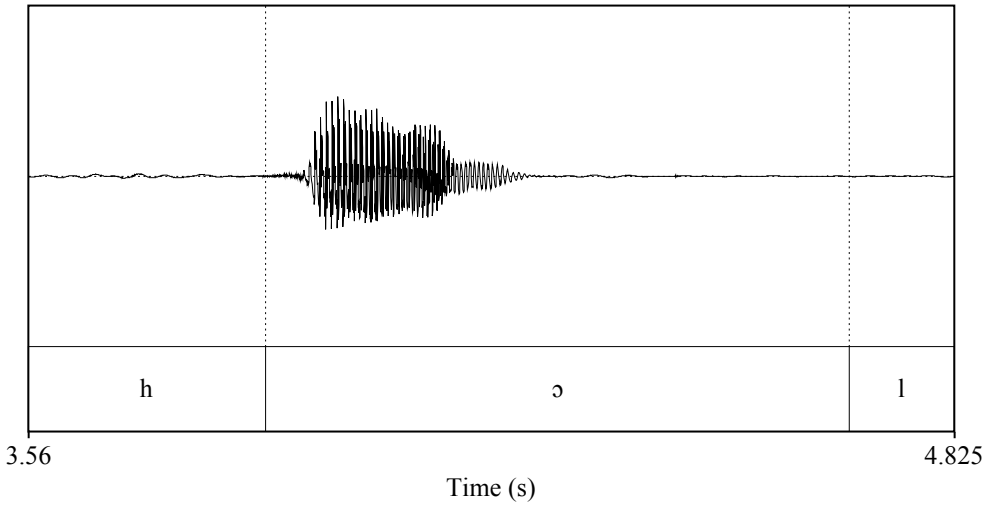
de ademgeluiden sterk genoeg zijn. De grens waaronder de stiltedetectie fragmenten als stil classificeert is in te stellen. Als deze grens echter te hoog ligt, worden ook sommige zachte fonetisch relevante geluiden, zoals de release van plosieven, weggefilterd.

Verder zijn stiltes en gevulde pauzes binnen een uiting problematisch. Gevulde pauzes, ook wel backchannel uitingen genoemd, zijn geluiden die gepaard gaan met denken. Een voorbeeld hiervan is de 'uh' in 'Ik uh weet het niet'. Deze kunnen aanzienlijk langer zijn dan andere fonemen in het geluidssignaal. Als deze gevulde pauzes en stiltes in de woordelijke transcriptie worden opgenomen, genereert de spraaksynthesizer een voorbeeldsignaal waarin de lengtes van de stiltes en gevulde pauzes niet langer is dan de andere fonemen. Hierdoor overschrijdt het verschil in lengte tussen het in het voorbeeldsignaal segment dat overeenkomt met de gevulde pauzes en de gevulde pauzes zelf de hellingsbeperking van het DTW algoritme. Als gevolg hiervan worden de segmenten voor en na de gevulde pauzes en stiltes uitgesmeerd over de stilte.

Ditzelfde probleem treedt op bij spraak met een overdreven prosodie ('Een grooote man').

Gerelateerd hieraan zijn de problemen die optreden bij spraak met lange pauzes tussen de woorden. Als de pauze van het TTS-systeem zijn ingesteld op 0, zoals is gebeurt bij alle experimenten die in deze scriptie zijn beschreven, zijn er geen pauzes in het voorbeeldsignaal die matchen met pauzes in het inputsignaal. Hierdoor worden woorden vaak wel redelijk goed geannoteerd, maar fonemen zelf niet, aangezien deze worden verdeeld over zowel het

woord als de stilte voor en na het woord. Figuur 4.6 geeft hier een voorbeeld van.



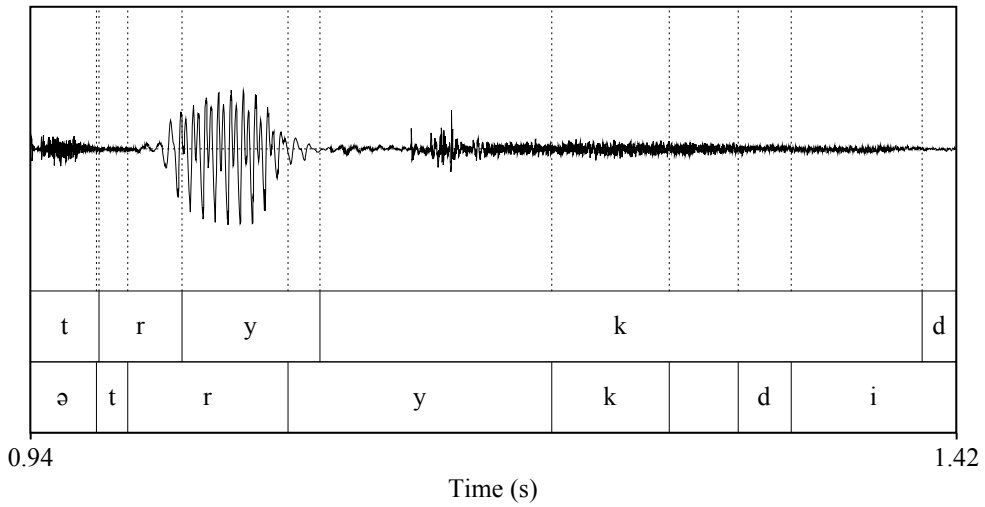
Figuur 4.6: Lange stiltes tussen woorden veroorzaken problemen voor de annotatie van fonemen, hoewel de woorden zelf nog steeds redelijk accuraat geannoteerd worden. In deze figuur is te zien dat de [h] en de [l] worden geannoteerd in stiltes. Dit figuur bevat het woord 'hol', afkomstig uit IFA-zin 2VY1A, uitgesproken door spreker M66O, geannoteerd met de aligned_corr methode.

Een laatste punt dat een enkele keer voorkwam in het IFA corpus, is dat sommige plosieven in lijzige spraak in het Nederlands een lange release vertonen. Dit veroorzaakt problemen die vergelijkbaar zijn met de problemen die worden veroorzaakt door stiltes en gevulde pauzes. Figuur 4.7 heeft hier een voorbeeld van.

Deze problemen worden veroorzaakt doordat niet aan de vierde voorwaarde uit paragraaf 1.2.1 wordt voldaan. Doordat het voorbeeldsignaal niet genoeg lijkt op het inputsignaal is een goede oplijning van het voorbeeldsignaal met het inputsignaal niet mogelijk, wat invloed heeft op de annotatie.

4.5.2 Inserties en deleties

Zoals uitgelegd in de inleiding is het van belang dat de text-naar-fonemen stap een foneemstring oplevert die overeenkomt met de fonemen aanwezig in het inputsignaal. Als dit niet het geval is, bevat de automatische annotatie andere segmenten dan idealiter het geval zou zijn. Dit uit zich voornamelijk in inserties en deleties. Door de insertie van een segment in de automatische annotatie, moet de 'correcte' grens tussen twee fonemen plaats maken voor twee incorrecte grenzen. De afwijkingen van de incorrecte grenzen ten opzichte van de correcte grens zijn min of meer gelijk, wat te zien is in figuren 4.3 en 4.4. In deze figuren is ook te zien dat de grenzen aan het begin en het eind van een of meerdere inserties sterker afwijkt van de handmatige annotatie, dan een grens tussen matchende fonemen.



Figuur 4.7: Een lange release van plosieven veroorzaakt problemen in de annotatie vergelijkbaar met de problemen veroorzaakt door stiltes in een signaal. Aan het einde van het woord 'truuk' wordt de release van de [k] langer aangehouden. Deze tekst is afkomstig uit IFA-zin K1FR2P, uitgesproken door spreker M40K, geannoteerd met de aligned methode. De onderste annotatie is de annotatie van het TTS-DTW systeem, de bovenste annotatie is de handmatige annotatie.

Insertie en deleties betekenen dat niet aan voorwaarde één uit paragraaf 1.2.1 wordt voldaan, omdat de annotatie van het TTS-DTW systeem in deze gevallen een ander aantal segmenten bevat dan de handmatige annotatie. Dit heeft twee gevolgen voor de automatische annotatie:

1. In het geval van inserties moet in de automatische annotatie plaats worden gemaakt voor geïnserteerde segmenten. Dit kan als gevolg hebben dat grenzen tussen andere segmenten moeten worden opgeschoven.

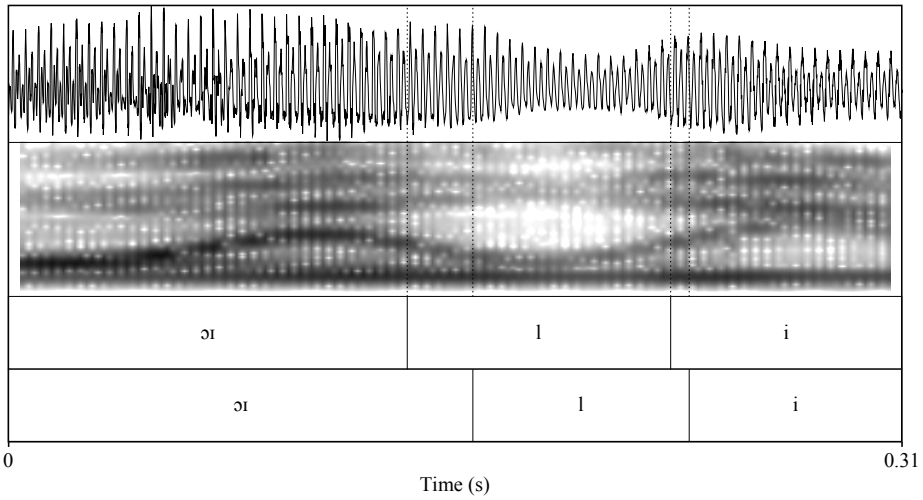
2. In het voorbeeldsignaal komen andere fonemen voor dan in het inputsignaal. Hierdoor wordt niet voldaan aan de vierde voorwaarde, doordat het voorbeeldsignaal nu minder op het inputsignaal lijkt.

In tabel 4.2 is te zien dat vooral veel inserties plaatsvinden, deleties komen minder vaak voor. Inserties zijn dus een groot probleem voor het TTS-DTW systeem.

4.5.3 Eigen annotatie spraaksynthesizer

Zoals besproken in sectie 3.1 worden overgangen tussen fonemen anders geannoteerd door de spraaksynthesizer dan in de corpora. Dit is terug te zien op plekken waar de oplijning van twee geluidssignalen verder goed is verlopen. Een voorbeeld hiervan is te zien in figuur 4.8. In dit figuur is te zien dat de grens tussen [l] en [i] door het TTS-DTW systeem wordt geplaatst op het moment dat de [i] stabiel is, terwijl de grens tussen deze fonemen in de

handmatige annotatie wordt geplaatst als de afstand tot de stabiele fase van beide fonemen gelijk is. Waarschijnlijk is dit een belangrijke oorzaak van de kleinere afwijkingen, hoewel dit niet op een geautomatiseerde wijze is na te gaan.



Figuur 4.8: Annotatie van het woord 'oily' door TIMIT en het TTS-DTW systeem. De handmatige TIMIT annotatie is de bovenste annotatie. Hier is te zien dat de grenzen tussen [ɔɪ] en [l] segmenten, en de [l] en [i] segmenten door het TTS-DTW systeem op de punten waar de formanttransitie (nagenoeg) voltooid is worden geplaatst, terwijl de handmatige annotatie midden in de transitie wordt geplaatst. Dit levert een afwijking van 23 ms op voor de eerste grens, en 7 ms voor de tweede grens.

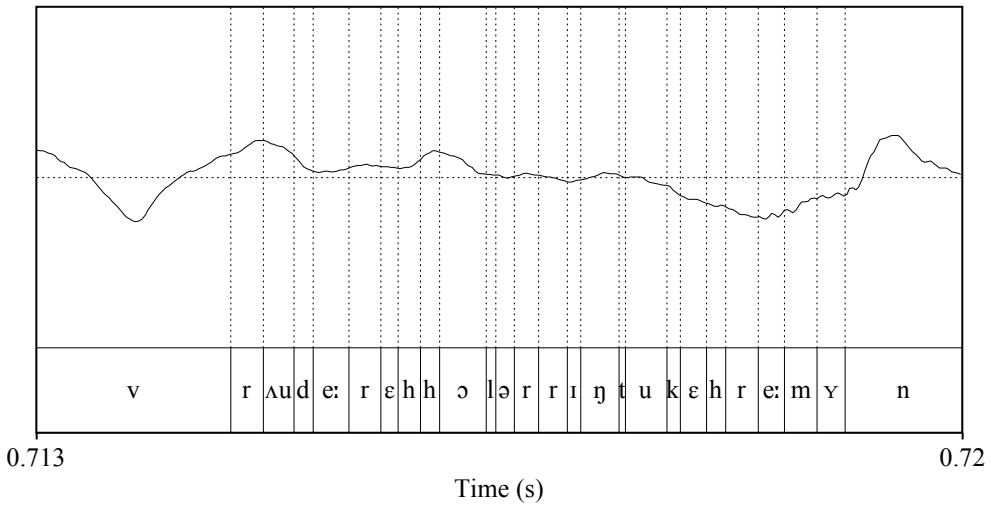
In dit geval wordt niet aan voorwaarde twee uit paragraaf 1.2.1 voldaan. Doordat de annotatie van het eigen signaal van de spraaksynthesizer systematisch afwijkt van de handmatige annotatie in de gebruikte corpora, komt de grens tussen fonemen ook in een perfecte oplijning van het voorbeeldsignaal met het inputsignaal op een andere plek te liggen dan in het corpus.

4.5.4 Verschillen tussen dialecten

Verschillen tussen dialecten kunnen niet direct verklaard worden uit verschillen in het fonetisch repertoire. Wellicht praten mensen in bepaalde dialecten langzamer, of laten ze meer pauzes vallen, of rekken ze sommige klanken meer uit. Dit laatste is zeker het geval voor de zuidelijke Amerikaans-Engelse dialecten, die ook significant slechter geannoteerd werden dan de andere Amerikaans-Engelse dialecten.

4.5.5 Problemen met DTW oplijning

In een aantal gevallen vertoont de oplijning van het inputsignaal met het voorbeeldsignaal nog problemen. Door een bug in het DTW algoritme worden alle grenzen op één punt in het inputsignaal gezet. Figuur 4.9 geeft hier een voorbeeld van.



Figuur 4.9: Een bug in het DTW algoritme. Door dat de oplijning een lang stuk uit het inputsignaal verwijdert en dit later weer invoegt, komen bijna alle grenzen op een plek in het inputsignaal te liggen, in dit geval in een gebied van minder dan 10 ms. In de figuur boven is een verdikking te zien tussen het [v]-segment en het [n]-segment, in de figuur onder wordt op deze verdikking ingezoomd. IFA-zin 2VY1A, uitgesproken door spreker M66O, geannoteerd met de aligned methode.

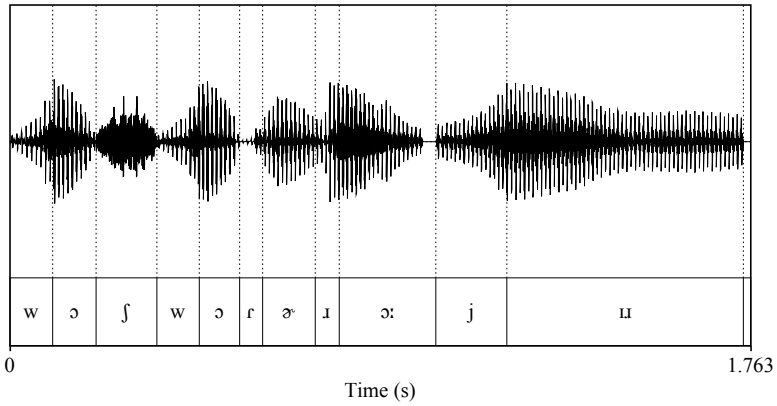
In dit geval wordt niet aan voorwaarde drie uit paragraaf 1.2.1 voldaan. Doordat er iets misgaat in het DTW algoritme, worden het inputsignaal en het voorbeeldsignaal niet goed met elkaar opgelijnd, waardoor de annotatie van het voorbeeldsignaal niet goed kan worden overgezet naar het inputsignaal.

4.5.6 Problemen met annotatie met foneemstring

In zowel het TIMIT corpus levert de aligned_phn methode slechtere resultaten. Dit is opmerkelijk, omdat deze methode was opgenomen om vast te stellen hoe goed het systeem werkt als geen text-naar-fonemenstap nodig is, en dus altijd de juiste fonemen worden gebruikt.

Hier zijn twee oorzaken voor te vinden. Ten eerste parseert de spraaksynthesizer foneemstrings niet altijd goed. In bepaalde gevallen worden twee afzonderlijke klinkers als één diftong of triftong geparseert. Soms levert dit ongeldige karakters op, waarna de spraaksynthesizer alleen de foneemstring tot aan het ongeldige karakter gebruikt voor het genereren van een voorbeeldsignaal.

Een voorbeeld hiervan is het woord ‘cereal’. De ARPAbet foneemstring hiervan is [s ih r iy el], in IPA is dit [sɪriḭ], met een syllabische [l]. In de Kirshenbaumtranscriptie wordt dit [sIri@L], waar @L staat voor een syllabische [l]. Om fonemen te vinden die uit meer dan éń ASCII teken bestaan, bijvoorbeeld diftongen, te vinden parseert de spraaksynthesizer een fonetische inputstring op een greedy manier: het kijkt bij elk teken vooruit om te zien



Figuur 4.10: Een synthese van de laatste woorden uit de TIMIT shibboleth-zin 'She had your dark suit in greasy wash water all year'. In deze figuur zijn de woorden 'wash water all year' gesynthetiseerd. Te zien is dat het [ɪɹ]-segment, wat overeenkomt met 'ear' uit 'year', aanzienlijk langer is dan alle andere segmenten.

of samenvoeging van het huidige teken met het volgende teken een geldig foneem oplevert. Verder stopt de spraaksynthesizer met het parseren van fonetische input zodra een ongeldig teken of het einde van de input wordt bereikt. De combinatie i@ is diftong die voor de spraaksynthesizer één foneem vormt. Hierdoor wordt [sIri@L] door de spraaksynthesizer geparseerd als [s], [I], [r], [i@], [L]. Het teken L vormt alleen een geldig foneem als het wordt voorafgegaan door @. Dit is nu niet meer het geval, waardoor de spraaksynthesizer stopt met parseren en de fonetische tekens volgend op @L negeert. In dit geval wordt niet aan voorwaarde één uit paragraaf 1.2.1 voldaan. De foneemstring die wordt gebruikt voor het genereren van een voorbeeldsignaal wijkt te sterk af van de foneemstring die aanwezig is in het inputsignaal.

Daarnaast is de fonemische input van de spraaksynthesizer meer gericht op het produceren van afzonderlijke woorden met een zeldzame uitspraak. de spraaksynthesizer ziet een foneemstring dan ook als één woord. Hierdoor wordt soms een vreemd geluidssignaal gevormd. Bovendien wordt het laatste foneem van spraak gesynthetiseerd uit een foneemstring soms op vreemde wijze gerekt. Hier staat een voorbeeld van in figuur 4.10. Hierdoor wordt niet aan voorwaarde vier uit paragraaf 1.2.1 voldaan. Doordat het voorbeeldsignaal niet genoeg lijkt op het inputsingaal is een goede oplijning van de twee signalen niet meer mogelijk.

Hoofdstuk 5

Discussie en suggesties

In deze scriptie is onderzocht of het TTS-DTW systeem handmatige annotatie kan bespoedigen en eventueel overbodig kan maken. Daarnaast is gekeken naar mogelijke oorzaken van problemen met het systeem. In dit hoofdstuk zal worden ingegaan op de in het vorige hoofdstuk besproken resultaten, en hoe de resultaten zich tot de vraagstelling van de scriptie verhouden. Verder zal een aantal suggesties ter verbetering van het TTS-DTW systeem in Praat worden geopperd.

5.1 Discussie

5.1.1 Vraagstelling

Vervanging handmatige annotatie

Uit de resultaten blijkt dat het TTS-DTW systeem in Praat annotaties oplevert die te sterk afwijken van handmatige annotatie om handmatige annotatie op dit moment volledig te vervangen. Met name wordt de afwijking tussen annotators zoals gerapporteerd door Cosi et al. (1991) niet bereikt.

Tijdswinst voor handmatige annotatie

Waarschijnlijk zal de automatische annotatie van het TTS-DTW systeem tijdswinst opleveren in het annotatieproces. Een vervolgstudie met een testgroep en controlegroep is nodig om te behaalde tijdswinst te kwantificeren. Van Son et al. (2001) rapporteren een factor 30-50 voor het corrigeren van een voorannotatie. Als deze factor ook toepasbaar blijkt als de annotatie van het TTS-DTW als voorannotatie wordt gebruikt, wordt een tijdswinst van een factor 2 geboekt ten opzichte van de door Jurafsky & Martin (2009) gerapporteerde factor 60-80. De kwaliteit van de voorannotatie is echter niet bekend, waardoor geen exacte vergelijking kan worden gemaakt het hier beschreven onderzoek.

Oorzaken

De oorzaken van problemen met automatische annotatie zijn grotendeels al besproken in het vorige hoofdstuk. Net als in Kominek et al. (2003) blijken niet-matchende stiltes een probleem te vormen voor automatische annotatie. Dit geldt echter voor alle afwijkingen tussen het voorbeeldsignaal en het inputsignaal. In tegenstelling tot Malfrère et al. (2003) vormen V>V overgangen in het Nederlandstalige IFA corpus een veel kleiner deel van de foneemovergangen dan in het Engelstalige TIMIT corpus. De V>V overgangen worden in beide corpora sterk afwijkend geannoteerd ten opzichte van de handmatige annotatie, maar vormen slechts een klein deel van het totale aantal foneemovergangen (6% in TIMIT, 0.1% in IFA), en zijn daarom waarschijnlijk geen grote oorzaak van afwijkingen in de automatische annotatie.

Daarnaast zijn alle gevonden problemen met automatische annotatie te wijten aan het schenden van één of meer van de voorwaarden gesteld in paragraaf 1.2.1.

5.1.2 Vergelijking eerdere onderzoeken en andere methode

Zoals door Kominek et al. (2003) al werd opgemerkt, veroorzaken niet matchende stiltes grote problemen voor de uitlijning. Ook in deze scriptie was dit het geval. Daarnaast bleken sterke afwijkingen in lengtes van spraakklanken grote problemen te veroorzaken. Deze twee problemen zijn beide aan voorwaarde vier uit sectie 1.2.1 verwant. Deze voorwaarde blijkt de belangrijkste te zijn.

De afwijking tussen grenzen tussen matchende segmenten in de handmatige en automatische annotaties verschilt niet sterk van eerdere onderzoeken naar TTS-DTW uitlijning, zoals Paulo & Oliveira (2003). De andere in de inleiding besproken onderzoeken naar TTS-DTW annotatie gaven betere resultaten dan de hier beschreven resultaten.

Een mogelijke verklaring hiervoor is dat alle andere implementaties van een TTS-DTW systeem gebruik maken van difoonsynthese, waardoor het verkregen spraaksignaal meer op menselijke spraak lijkt. Bovendien vinden zijn coarticulatie-effecten hierdoor natuurlijker verwerkt in het gesynthetiseerde signaal, waardoor grenzen tussen fonemen in het voorbeeldsignaal meer lijken op grenzen tussen fonemen zoals die voorkomen in natuurlijke spraak.

Het systeem laat slechtere resultaten zien dan modernere methodes besproken in de inleiding, zoals Keshet et al. (2007) en Jakovljević et al. (2012).

5.2 Suggesties

5.2.1 Stiltes en gevulde pauzes

Een oplossing voor ademgeluiden voor en na uitingen is vrij simpel te implementeren. Praat bevat een mogelijkheid om alle stiltes in een signaal te vinden en deze te annoteren. Als een

niet-stil signaal zich tussen twee stille signalen bevindt, en geen stemhebbende kenmerken heeft, is dit zeer waarschijnlijk een ademgeluid. Belangrijk is hier wel dat de minimale lengte van stille segmenten niet te laag wordt ingesteld, omdat dan ook de release van twee opeenvolgende plosieven als stil kan worden bestempeld.

Een betere oplossing voor alle problemen die in de sectie Problemen met stiltes en gevulde pauzes zijn besproken, zou zijn om voordat het DTW algoritme wordt uitgevoerd, eerst alle plekken in het signaal te vinden die fonetisch voor lange tijd zeer weinig verandering vertonen, en deze terug te brengen tot een maximale lengte. Dit zou dan naast stiltes ook langere gevulde pauzes op kunnen vangen.

5.2.2 Inserties en deleties

Zoals te zien in figuren 4.4 en 4.5 hebben inserties en deleties een negatieve invloed op de kwaliteit van de uitlijning. Bovendien leveren deze extra werk op bij een handmatige nacontrole: het uit een textgrid verwijderen van geïnserteerd kost op dit moment 4 operaties: het selecteren van het ingevoegde interval, het verwijderen van de tekst uit het interval, het selecteren van de grens van het interval die moet worden verwijderd en het verwijderen van de grens.

Makkelijker zou zijn als het interval met één operatie kon worden verwijderd. Dit roept echter de vraag op waar de grens tussen de aan het geïnserteerde interval grenzende segmenten moet komen te liggen. Een mogelijke oplossing hiervoor zou kunnen liggen in het inbouwen van drie functies: verwijder interval, maak rechtergrens aan linkergrens vast; verwijder interval, maak linkergrens aan rechtergrens vast; verwijder interval, verbind linkergrens en rechtergrens in het midden. Hiermee zou de nacontrole van de output van het TTS-DTW algoritme kunnen worden versneld.

5.2.3 Foneemstring

Wellicht kan een onderzoek waarbij de foneemstring wordt opgesplitst in losse woorden een beter licht werpen op de mogelijkheid van een foneemstring als input. Binnen deze scriptie was dit niet meer mogelijk.

Daarnaast zou een foneemstring kunnen worden gegenereerd die meer lijkt op de foneemstring in het gesproken signaal door gebruikers een keuze te laten maken tussen verschillende mogelijke fonetische representaties van een tekstuele inputzin. Paulo & Oliveira (2005) stellen een systeem voor dat waarschijnlijke uitspraken van woorden genereert, gebaseerd op gewogen finite state transducers.

5.2.4 Eigen annotatie spraaksynthesizer

De grenzen tussen fonemen zoals in figuur 3.1 worden op een vreemde plek geplaatst. Zoals eerder opgemerkt is de plaatsing van een grens tussen fonemen arbitrair, maar kan het wel belangrijk zijn dat de plaatsing consistent gebeurt. Als het TTS-DTW systeem de grenzen in het inputsignaal consistent plaatst, ook als dit afwijkt van een handmatige annotatie, zou dit al een goede prestatie zijn (van Son, persoonlijke communicatie). Dat de annotatieconventies van de gebruikte spraaksynthesizer niet precies overeenkomen met een handmatige annotatie is daarom vooral nadelig voor vergelijking met een handmatig geannoteerd corpus.

Wel zou het interessant kunnen zijn om gebruikers van het TTS-DTW systeem de mogelijkheid te geven annotatieconventies te specificeren. Hoewel de precieze plaatsing van grenzen tussen fonemen alleen kan worden beïnvloed door de broncode van eSpeak aan te passen, is het daarom wellicht waardevol om hier naar te kijken.

5.2.5 Oplijning input- en voorbeelsignaal

In de implementatie van het DTW algoritme in Praat worden alleen MFCC's gebruikt in de feature vectors. In Paulo & Oliveira (2003) wordt een aangepast TTS-DTW systeem besproken. Hierin worden, afhankelijk van de foneemovergang in het voorbeeldsignaal, verschillende feature vectors gebruikt. Hiermee worden betere resultaten geboekt ten opzichte van handmatige annotatie dan met enkel MFCC's als feature vectors.

5.3 Laatste opmerking

Het TTS-DTW systeem in Praat was tijdens het schrijven dezes nog in ontwikkeling. Het onderzoeken van het systeem is daarom gepaard gegaan met het zoeken naar en rapporteren van weeffouten en oneffenheden in het systeem. De verschillende gevonden bugs zijn echter niet in deze scriptie opgenomen.

Hoofdstuk 6

Conclusie

Zoals besproken in de tekst werkt het TTS-DTW systeem redelijk goed: in het beste geval, automatische annotatie op het TIMIT corpus met de aligned methode, is 44,34% van alle grenzen tussen fonemen geplaatst op een <20 ms afstand van de grens in een handmatige annotatie.

Uit de resultaten van de experimenten blijkt dat het TTS-DTW systeem een begin van een annotatie kan zijn. Een volledig correcte annotatie kan echter alleen onder zeer goede condities worden verkregen. Hiervoor is een spreker/-ster nodig die alle fonemen in elk woord realiseert, geen pauzes laat vallen tussen woorden, geen klanken te lang aanhoudt en een niet te uitbundige prosodie vertoont. Op deze wijze spreken is echter onnatuurlijk, en zal niet in veel opgenomen spraak kunnen worden teruggezien.

Om te kunnen kwantificeren hoeveel tijdswinst het TTS-DTW systeem in Praat oplevert, zou een onderzoekje gedaan kunnen worden waarin proefpersonen een TTS-DTW annotatie voor een signaal corrigeren, en een controlegroep datzelfde signaal zonder hulp van TTS-DTW annoteert.

Om betere resultaten te boeken in de toekomst moet gewerkt worden aan het derde of het vierde punt uit paragraaf 1.2.1: factoren die mogelijk verschillen tussen het voorbeeldsignaal en het inputsignaal veroorzaken, zoals stiltes en het lang aanhouden van fonemen, kunnen wellicht vooraf worden opgespoord. Eén mogelijkheid daarvoor is het vinden van stiltes en lange fonemen in het spraaksignaal en deze terugbrengen tot een maximale lengte voordat het DTW algoritme wordt uitgevoerd. Een andere mogelijkheid is het lokaal loslaten van de hellingsbeperking in het DTW algoritme als er punten zijn in het kortste legale pad door de afstandsmatrix waar de afstand tussen feature vectors uit het bronsignaal en het doelsignaal te veel van elkaar verschillen.

Op dit moment levert TTS-DTW met foneemstring als input nog geen goede resultaten. Onderzoek met een foneemstring opgesplitst in woorden zou wellicht betere resultaten kunnen opleveren. De andere problemen die zijn beschreven in paragraaf 4.5.6 zijn daarmee nog niet opgelost.

Nawoord

Graag zou ik mijn David Weenink bedanken voor zijn begeleiding bij het schrijven van deze scriptie. Zonder zijn kennis, hulp en commentaar zou deze scriptie niet mogelijk zijn geweest.

Bibliografie

Adell, Jordi, Antonio Bonafonte, Jon Ander Gómez & María José Castro (2005), Comparative study of automatic phone segmentation methods for TTS, in *Proceedings of of IEEE International Conference on Acoustics, Speech and Signal Processing, Philadelphia, USA*, pag. 309–312.

Black, Alan W. & Kevin A. Lenzo (2007), *Building Synthetic Voices (Festival Documentation)*, version 27-01-2007, Retrieved 28-03-2012 from `http://www.festvox.org/festvox/book1.html`.

Boersma, Paul & David Weenink (2012), Praat: Doing Phonetics by Computer [Computer Program], version 5.3.16, Retrieved 31-05-2012 from `http://www.praat.org/`.

Cosi, Piero, Daniele Falavigna & Maurizio Omolog (1991), A Preliminary Statistical Evaluation of Manual and Automatic Segmentation Discrepancies, in *Proceedings of Eurospeech 1991, Genova, Italy*, pag. 693–696.

Davis, S.B. & P. Mermelstein (1980), Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences., *IEEE Transactions on ASSP*, vol. 28:pag. 357–366.

Duddington, Jonathan (2011), eSpeak text to speech [Computer Program], version 1.46, Retrieved 12-01-2012 from `http://espeak.sourceforge.com/`.

Dutoit, Thierry (1999), MBROLIGN [Computer Program], version 1, Retrieved 18-03-2012 from `http://tcts.fpms.ac.be/synthesis/mbrolign/`.

Dutoit, Thierry, Vincent Pagel, F. Bataille & Olivier van der Vreken (1996), The MBROLA Project: Towards a Set of High-Quality Speech Synthesizers Free of Use for Non-Commercial Purposes, in *Proceedings of ICSLP'96, Philadelphia, vol. 3*, pag. 1393–1396.

Holmes, John & Wendy Holmes (2001), *Speech Synthesis and Recognition*, Taylor & Francis, New York, NY, 2e druk.

Horák, Petr (2001), Automatic Speech Segmentation Based on Alignment with a Text-to-Speech System, in E. Keller, G. Bailly, A. Monahan, Terken J. & M. Huckwale, red., *Improvements in Speech Synthesis*, hfdst. 33, pag. 331–340, John Wiley and Sons Ltd.

Jakovljević, Nikša, Dragiša Mišković, Darko Pekar, Milan Sečujski & Vlado Delić (2012), Automatic Phonetic Segmentation for a Speech Corpus of Hebrew, *Infoteh-Jahorina*, vol. 11:pag. 742–745.

Jurafsky, Daniel & James H. Martin (2009), *Speech and Language Processing*, Pearson Education, Inc., Upper Saddle River, NJ., 2e druk.

Keshet, Joseph, Shai Shalev-Shwartz, Yoram Singer & Dan Chazan (2007), A Large Margin Algorithm for Speech-to-Phoneme and Music-to-Score Alignment, *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15(8):pag. 2373–2382.

Kirshenbaum, Evan (2001), *Representing IPA phonetics in ASCII*, version 6-9-2011, Retrieved 28-03-2012 from `http://www.kirshenbaum.net/IPA/ascii-ipa.pdf`.

Kominek, John, Christina Bennet & Alan W. Black (2003), Evaluating and Correcting Phoneme Segmentation for Unit Selection Synthesis, in *Proceedings of Eurospeech 2003, Geneva, Switzerland*, pag. 313–316.

Lamel, L.F., R. H. Kassel & S. Seneff (1986), Speech Database Development: Design and Analysis of the Acoustic-Phonetic Corpus.

Malfrère, Fabrice, Olivier Deroo, Thierry Dutoit & Christophe Ris (2003), Phonetic Alignment: Speech Synthesis-Based vs. Viterbi-Based, *Speech Communication*, vol. 40:pag. 503–515.

Malfrère, Fabrice & Thierry Dutoit (1997), High Quality Speech Synthesis for Phonetic Speech Segmentation, in *Proceedings of Eurospeech 1997, Rhodes, Greece*, pag. 2631–2634.

Paulo, Sergío & Luís C. Oliveira (2003), DTW-based Phonetic Alignment Using Multiple Acoustic Features, in *Proceedings of Eurospeech 2003, Geneva, Switzerland*, pag. 309–312.

Paulo, Sérgio & Luís C. Oliveira (2005), Generation of Word Alternative Pronunciations Using Weighted Finite State Transducers, in *Proceedings of Interspeech 2005, Lisbon, Portugal*, pag. 1157–1160.

R Development Core Team (2012), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0.

Sakoe, Hiroaki & Seibi Chiba (1978), Dynamic Programming Algorithm Optimization for Spoken Word Recognition, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26(1):pag. 43–49.

Seltzer, Michael & Rita Singh (2012), *Sphinx3 Training Manual*, retrieved 29-04-2012 from `http://www.speech.cs.cmu.edu/sphinxman/`.

van Son, R.J.J.H., Diana Binnenpoorte, Henk van den Heuvel & Louis C.W. Pols (2001), The IFA Corpus: a Phonemically Segmented Dutch "Open Source"Speech Database, in *Proceedings of Eurospeech 2001, Aalborg, Denmark, vol. 3*, pag. 2051–2054.

Taylor, Paul (2009), *Text-to-Speech Synthesis*, Cambridge University Press, Cambridge, UK, 1e druk.

Taylor, Paul A, Alan Black & Richard Caley (1998), The Architecture of the Festival Speech Synthesis System, in *The Third ESCA Workshop in Speech Synthesis*, pag. 147–151, Jenolan Caves, Australia.

Weenink, David (unpublished), *Speech Signal Processing by Praat*.

Wells, J.C. (1997), SAMPA computer readable phonetic alphabet, in D. Gibbon, R. Moore & R. Winski, red., *Handbook of Standards and Resources for Spoken Language Systems*, hfdst. IV, section B, Mouton de Gruyter, Berlin and New York.

Zwicker, E. (1961), Subdivision of the audible frequency range into critical bands (Frequenzgruppen), *Journal of the Acoustical Society of America, in: Holmes & Holmes (2001)*, vol. 33:pag. 248.

Appendix A: Transcriptie-aanpassingen

In deze sectie wordt besproken welke aanpassingen nodig waren om de handmatige annotaties van corpora te vergelijken met de output van het TTS-DTW systeem.

TIMIT

TIMIT annoteert de stiltes en de release van plosieven apart. In de annotatie van de spraaksynthesizer gebeurt dit niet. Zoals eerder besproken annoteert de spraaksynthesizer de stiltes van stemloze plosieven bij het voorgaande foneem, terwijl de stilte van stemhebbende plosieven bij het volgende foneem wordt geannoteerd. Deze discrepantie is voor TIMIT opgelost door alle segmenten met plosiefstiltes die worden gevolgd door een segment met een stemhebbende plosief samen te voegen met het volgende segment, en alle segmenten met plosiefstiltes die worden gevolgd door een stemloze plosief samen te voegen met het vorige segment.

Daarnaast moesten enkele fonemen worden samengevoegd, omdat deze door de spraaksynthesizer als een segment werden geannoteerd. Voor alle overgebleven fonemen was een directe vertaling naar de Kirshenbaumtranscriptie mogelijk. Voor het transformeren van de TIMIT annotatie naar de Kirshenbaumannotatie van de spraaksynthesizer wordt het onderstaande algoritme op elk TIMIT annotatiebestand uitgevoerd. Intervallen worden vertaald door de tekst in het interval op te zoeken in ARPAbet kolom van de tabel die onder het algoritme staat, en de tekst in het interval te vervangen door de tekst in de Kirshenbaum kolom.

```
index = 1 // eerste interval
zolang i <= aantal intervallen
   ga_verder = FALSE
   als interval[i] een plosiefsluiting is:
      als interval[i+1] een stemloze plosief is:
         verwijder tekst uit interval[i]
```

```
        voeg interval[i] samen met interval[i-1]
        ga_verder = TRUE
    anders als interval[i+1] een stemhebbende plosief is:
        verwijder tekst uit interval[i]
        voeg interval[i] samen met interval[i+1]
        ga_verder = TRUE
    anders:
        ga_verder = TRUE
anders als interval[i] en interval[i+1] samen een foneem vormen:
    voeg intervallen samen
anders:
    ga_verder = TRUE
als ga_verder:
    vertaal interval[i]
    i = i + 1
```

IFA

De omschrijving van de SAMPA transcriptie van het IFA corpus naar de Kirshenbaumtranscriptie is simpeler te implementeren. De SAMPA en Kirshenbaumtranscripties hebben veel overeenkomsten, en waar de beide transcripties van elkaar verschillen is altijd een één-op-één vertaling mogelijk. De fonemen die van de SAMPA transcriptie naar de Kirshenbaumtranscriptie moesten worden vertaald staan in onderstaande tabel. Verder gebruikt de SAMPA transcriptie het teken '=' om klanken aan te geven dat fonemen niet gescheiden kunnen worden. Dit is niet mogelijk in de Kirshenbaumtranscriptie, en alle voorkomens van '=' zijn daarom verwijderd.

Tabel 6.1: De vertaalsleutel voor het vertalen van een annotatie in de ARPAbet transcriptie in een annotatie in de Kirshenbaum transcriptie

ARPAbet	Kirshenbaum		ARPAbet	Kirshenbaum		ARPAbet	Kirshenbaum
b	b		l	l		uhr	U@
d	d		r	r		ayaxr	aI3
g	g		w	w		awaxr	aU@
p	p		y	j		ax	@
t	t		el	@L		ix	I2
k	k		iy	i		ax-h	@
q	_		ih	I		h#	_
dx	t#		eh	E		#h	_
jh	dZ		ey	eI		pau	_
ch	tS		ae	a		_	_
s	s		aw	aU			
sh	S		ay	aI			
z	z		oy	OI			
zh	Z		ow	oU			
f	f		uh	U			
th	T		uw	u:			
v	v		ux	u:			
dh	D		aa	A			
hh	h		ah	V			
hv	h		ao	O:			
m	m		aar	A@			
n	n		aer	e@			
ng	N		ihr	i@r			
em	m		er	3:			
en	n		axr	3			
eng	N		oar	O@			
nx	n		aor	O@			

Tabel 6.2: Vertaalsleutel voor symbolen die verschillen in de SAMPA en Kirshenbaumtranscriptie.

SAMPA	Kirshenbaum		SAMPA	Kirshenbaum
*	_		X	Q
0	O		G	Q
1	I		9	Wy
Y	8		D	d
E+	EI		F	f
9+	Wy		H	h
w	v#		J	j
2	Y:		L	l
O+	VU		b	b
x	Q		o+	VU

Appendix B: Editkostenmatrices

De editkostenmatrices zijn te groot om op een A4-pagina leesbaar afgedrukt te kunnen worden. Daarom zijn de scripts die gebruikt zijn om de editkostenmatrices te maken opgenomen in Appendix C: Scripts.

Appendix C: Scripts

In deze scriptie zijn enkele scripts gebruikt. In deze appendix wordt beschreven hoe de scripts zijn gebruikt en in welke volgorde. Verder wordt uitgelegd waar op moet worden gelet als de scripts worden gebruikt om de experimenten te reproduceren. De scripts zijn, vanwege de omvang van het totaal aantal scripts, niet in deze appendix opgenomen, maar kunnen op aanvraag worden opgestuurd.

Een aantal scripts vereist een *NIX systeem vanwege systeemcommando's die in de scripts worden aangeroepen. De scripts die een *NIX systeem vereisen zijn gemarkeerd met een *.

Bestandsstructuren

Vanwege verschillende bestandsstructuren en afwijkende conventies voor het benoemen van bestanden, zijn voor de IFA en TIMIT corpora verschillende scripts gebruikt. De scripts zijn geschreven voor het TIMIT corpus, en later aangepast voor het IFA corpus.

De bestanden in het TIMIT corpus staan in de volgende mapstructuur:

```
{HOOFDMAP}/timit/[train|test]/dr[1-8]/{spreker}/{zin}.[wav|phn|wrd|txt]
```

In elke spreker-map zijn de submappen TextGrid en Table aangemaakt. Deze bevatten respectievelijk de automatische annotaties (alle drie de methodes), en de output van het string alignment algoritme op deze tabellen.

Doordat de bestanden in het IFA corpus alle een unieke bestandsnaam hebben (in tegenstelling tot het TIMIT corpus), kon het IFA corpus in een map worden samengevoegd. De IFA hoofdmap bevat de volgende submappen TG en AIFC. De AIFC submap bevat alle geluidsdata, en de TG submap de verschillende annotatiebestanden. De TG submap bevat de volgende submappen:

- orig: deze map bevat de originele annotaties van het IFA corpus
- orig_edited: deze map bevat de annotaties die zijn aangepast voor de Kirshenbaum transcriptie

- synth_out: deze map bevat de annotaties die het TTS-DTW systeem heeft voortgebracht
- Tables: deze map bevat de tabellen die het string-alignment algoritme heeft gegenereerd

Scripts

Deze sectie geeft een korte beschrijving van elk script. Tabel 1 geeft aan welke TIMIT scripts overeenkomen met welke IFA scripts, en in welke volgorde de scripts moeten worden uitgevoerd.

Tabel 6.3: De verschillende scripts voor de TIMIT en IFA corpora. De scripts staan in de volgorde waarin ze worden uitgevoerd. Scriptnamen gevolgd door een * vereisen een *NIX systeem, scriptnamen gevolgd door een [1] worden aangeroepen door het script all_files.praat.

TIMIT	**IFA**
TIMIT_convert_transcription.py*	IFA_convert_transcription.py*
all_files.praat	-
do_proc_make_annotation.praat*[1]	maak_synth_TGs.praat*
check_files.praat*[1]	check_files.praat*
maak_editkostenmatrix.praat	fonemen.praat; maak_editkostenmatrix.praat
TIMIT_string_alignment.py*	IFA_string_alignment.py
do_proc_read_tables_boundary.praat[1]	find_displacements_new_3.praat
do_proc_one_giant_table.praat[1]	compile_tables.praat
insertie_effecten.praat	insertie_effecten.praat

In de beschrijving van elk script hieronder worden de namen van de TIMIT scripts aangehouden.

TIMIT_convert_transcription.py

Dit script converteert de Arpabet transcriptie van het TIMIT corpus en de SAMPA transcriptie van het IFA corpus naar een Kirshenbaum transcriptie. De uitvoer is een TextGrid met een woordelijke transcriptie, een fonetische transcriptie en een samengevoegde fonetische transcriptie van de uiting.

all_files.praat

Dit script, welke alleen wordt gebruikt voor het TIMIT corpus, roept voor elk bestand van een bepaald type in alle spreker-mappen een procedure do_proc aan. De do_proc procedure heeft twee argumenten: een map, en een bestandsnaam zonder extensie. De do_proc procedure staat moet in een ander script staan dat is geïnclude in het all_files script.

do_proc_make_annotation.praat

Dit script voert het TTS-DTW algoritme uit op elk bestand in de corpora. Bestanden waarvoor geen transcriptie is gelukt worden voor het TIMIT corpus naar de console van Praat geschreven, en voor het IFA corpus in een tabel opgeslagen.

check_files.praat

Dit script checkt of de uitvoer van het TTS-DTW algoritme kan worden gelezen door Praat. Indien dit niet het geval is wordt de betreffende TextGrid verwijderd. De bestandsnaam van het onleesbare bestand wordt voor het TIMIT corpus naar de Praat-console geschreven, en voor het IFA corpus opgeslagen in een aparte tabel.

maak_editkostenmatrix.praat

Dit script maakt een editkostenmatrix voor het betreffende corpus. De editkostenmatrix wordt naar het Praat-console geschreven in de Python syntax voor een dictionary (Hashtable), zodat deze kan worden gebruikt in het String-alignment algoritme. De fonemen zijn voor het IFA corpus in een apart bestand opgeslagen.

TIMIT_string_alignment.py

Voert string alignment uit op de uitvoer van het TTS-DTW algoritme en de handmatige annotatie. De uitvoer wordt opgeslagen in een door Praat leesbare tabel.

do_proc_read_tables_boundary.praat

Vindt grenzen tussen matchende segmenten in de uitvoer van het TTS-DTW algoritme.

do_proc_one_giant_table.praat

Voegt alle string-alignment tabellen samen met toevoeging van wat meta-informatie. Deze tabellen zijn handmatig gesplitst op methode (aligned, aligned_corr of aligned_phn). De gesplitste tabellen zijn opgeslagen als bestanden met comma-separated-values om door R gelezen te kunnen worden.

insertie_effecten.praat

Berekend de afwijking ten opzichte van de handmatige annotatie als gevolg van inserties. Ook deze tabellen zijn als bestanden met comma-separated-values opgeslagen om door R gelezen te kunnen worden.