

Spraakanalyse voor presentaties

Auteur: Bas de Haan

Informatica, De Haagse Hogeschool, Den Haag

Studentnummer: 11098511

Begindatum: 01-05-2015

Einddatum: 05-10-2015

Bedrijf: Ordina, Nieuwegein

Examinatoren HHS: Dhr. E.M. van Doorn

Dhr. G.M. Tuk

Begeleider Ordina: Dhr. F. Verbruggen

Opdrachtgever: Dhr. R. van Tilborg

Voorwoord

Als afronding van mijn HBO Informatica opleiding aan De Haagse Hogeschool heb ik mijn afstuderen gedaan bij Ordina, te Nieuwegein. Naast de ontwikkelde software is dit verslag daar het resultaat van.

Bij Ordina hadden ze enkele opdrachten klaarliggen voor afstudeerders, deze opdracht leek mij de interessantste.

Voor de begeleiding wil ik graag de heren Verbruggen, van Doorn en Tuk bedanken. Daarnaast wil ik de heer Weidner bedanken voor zijn inzet als manager en aanspreekpunt. Ten slotte ook de heer van Tilborg, die heeft opgetreden als opdrachtgever.

Inhoudsopgave

Voorwoord	ii
Inhoudsopgave	iii
1. Inleiding	1
2. Organisatie	2
3. De opdracht	4
3.1. Uitgangssituatie	4
3.2. Probleemstelling	4
3.3. Doel van de opdracht	4
3.3.1. Wat is een presentatie?	4
3.3.2. Potentieel probleem	5
3.3.3. Mogelijkheden en oplossing	5
4. Beschrijving van de werkzaamheden	6
4.1. Timebox 1 – Plan van aanpak en onderzoek	6
4.1.1. Plan van aanpak	6
4.1.2. Achterhalen requirements	7
4.1.3. Onderzoek open-source Java spraakherkenningslibraries	8
4.2. Timebox 2 – Ontwerp en aanpassing werkwijze	12
4.2.1. UML ontwerp	12
4.2.2. Werkwijze	13
4.3. Timebox 3 – Eerste implementaties API's en aspecten	14
4.3.1. Planning	14
4.3.2. Testbestanden	14
4.3.3. Implementatie API's	14
4.3.4. Standaardisatie van de output	15
4.3.5. Transcripties	16
4.3.6. Aspecten	17
4.3.7. Implementatie alternatievenanalyse	20
4.3.8. Niet-presentaties	21
4.3.9. Voortschrijdend inzicht aanpak	21
4.4. Timebox 4 – Stille- en snelheidsanalyse	23
4.4.1. Planning	23
4.4.2. Document updates	24
4.4.3. Tussentijds uitvoeren analyse	25

4.4.4.	Implementatie aspecten	26
4.4.5.	Tests.....	29
4.4.6.	Bespreking met begeleider.....	31
4.4.7.	Versnellen.....	32
4.4.8.	Becijferen.....	33
4.4.9.	GUI	33
4.4.10.	Herocoderen audio.....	33
4.4.11.	UML update	34
4.4.12.	YouTube API	34
4.5.	Timebox 5 – Testen, refactoren en paralleliseren.....	35
4.5.1.	Planning	35
4.5.2.	Analyses tien bestanden.....	35
4.5.3.	Verschillen in resultaat van zelfde bestand.....	36
4.5.4.	Bespreking met opdrachtgever	37
4.5.5.	Bespreking met begeleider.....	38
4.5.6.	Parallel analyseren	38
4.5.7.	UML update	39
4.6.	Timebox 6 – YouTube API.....	40
4.6.1.	Planning	40
4.6.2.	Ted Talks download.....	40
4.6.3.	YouTube API	40
4.6.4.	UML update	41
4.7.	Timebox 7 – Afronding analyses, voorbereidingen machine learning	42
4.7.1.	Analyses afronden	42
4.7.2.	Voorbereidingen voor Machine Learning.....	42
4.8.	Timebox 8 – CSV-export en becijfering YouTube	43
4.8.1.	Planning	43
4.8.2.	CSV	43
4.8.3.	Categoriseren en/of becijferen YouTube-data.....	43
4.9.	Timebox 9 – Live presentaties.....	44
4.9.1.	Planning	44
4.9.2.	Live presentaties.....	44
4.10.	Timebox 10 – Machine learning en API-documentatie	45
4.10.1.	Planning	45

4.10.2.	Machine learning.....	45
4.10.3.	UML-diagram.....	46
4.10.4.	API-documentatie.....	46
5.	Afwijkingen ten opzichte van het afstudeerplan	47
6.	Niet-opgeleverde producten.....	48
7.	Evaluatie	49
7.1.	Aanpak.....	49
7.2.	Tussenproducten.....	49
8.	Te demonstreren competenties en wijze waarop	50
9.	Geraadpleegde literatuur en andere bronnen.....	52
10.	Afkortingenlijst en noodzakelijke, niet algemeen toegankelijke achtergrondinformatie	52
11.	Bijlagen	53
	Bijlage A – Plan van Aanpak	53
	Bijlage B – Onderzoek API's en libraries.....	59
	Bijlage C – Zoeken naar instellingen voor consistente resultaten	67
	Bijlage D – Transcripties	71
	Bijlage E - Tests	79
	Bijlage F – Ontwerp	84
	Bijlage G – Resultaten 10 bestanden.....	94
	Bijlage H – Requirements	96
	Bijlage I – API-documentatie	97
	Bijlage J – Afstudeerplan	98
	Bijlage K – Tussentijdse beoordelingen.....	102
	Bijlage L – Resultaten machine learning.....	107

1. Inleiding

Ik heb mijn afstuderen gedaan bij Ordina te Nieuwegein. Dit was van 1 mei tot 5 oktober 2015.

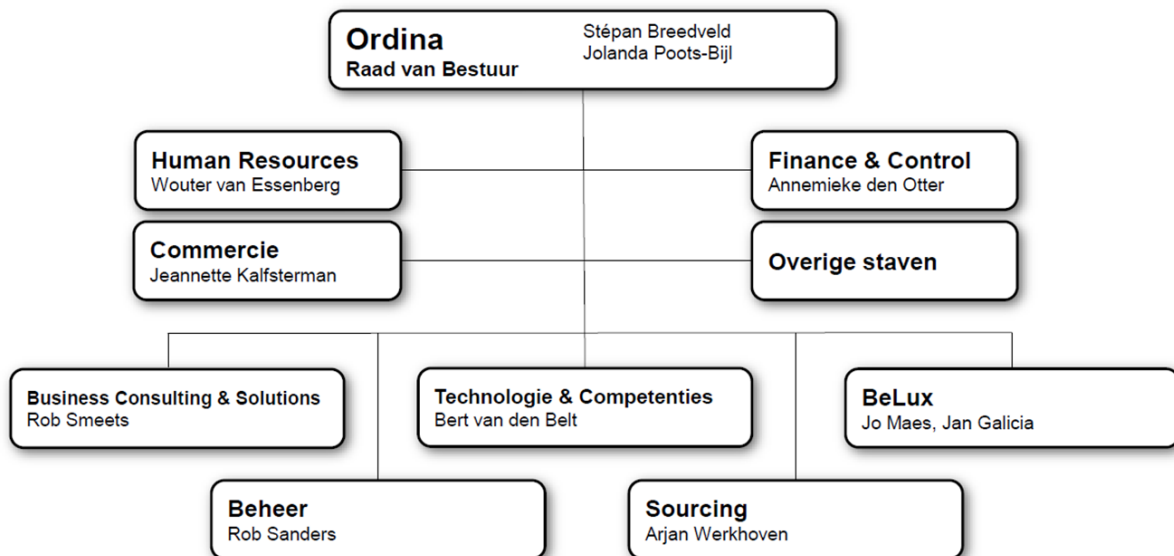
Ordina is een relatief groot softwareontwikkelings- en detachingsbedrijf, werkzaam in de gehele Benelux. Ik was daar zelfstandig werkzaam als software developer. Het project waar ik aan werkte hield kortgezegd in dat ik software ontwikkelde die op basis van alleen de audio van een presentatie de presentatie kan beoordelen.

In dit document zal ik inzicht geven in mijn werkzaamheden.

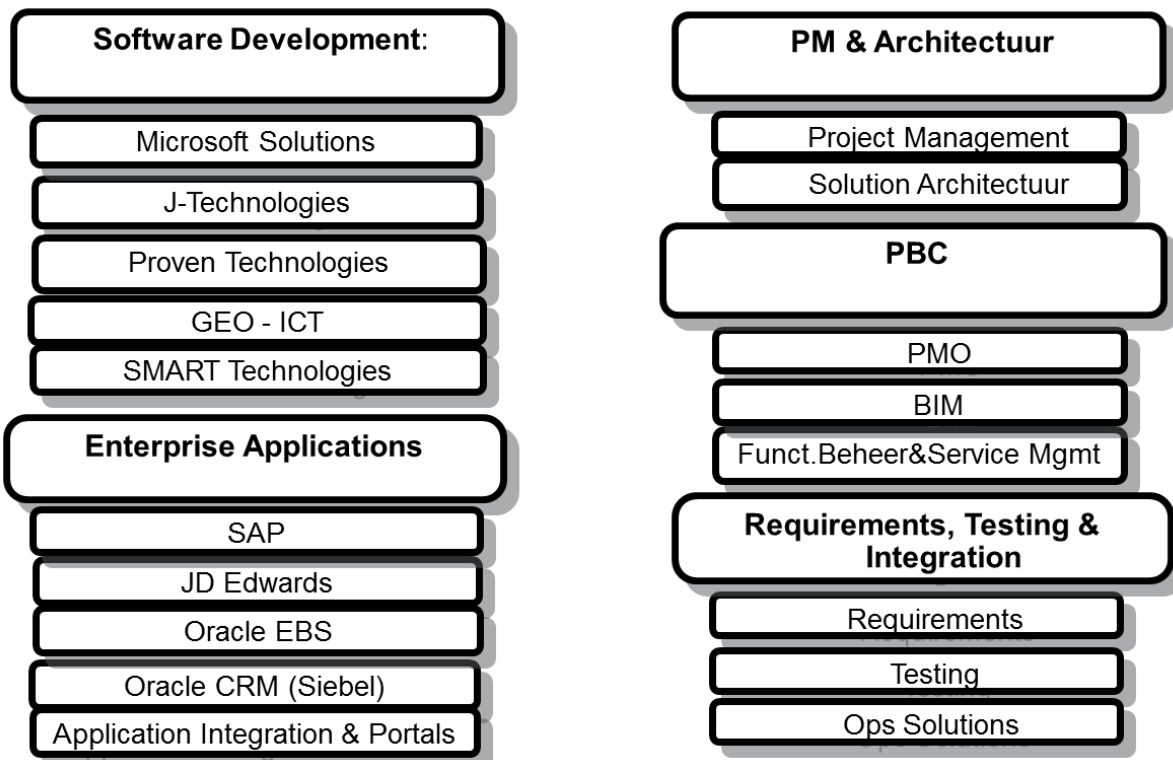
2. Organisatie

Ordina is een bedrijf uit 1983, heeft zo'n 2.500 medewerkers. Ordina ontwikkelt business- en IT-oplossingen voor verschillende markten. Deze markten zijn de industrie, de financiële wereld, de overheid en de zorg. Naast het ontwikkelen van business- en IT-oplossingen worden er ook trainingen georganiseerd door Ordina, waaronder presentatietrainingen. De opdracht heeft betrekking op de presentatietrainingen en wordt uitgevoerd binnen de J-technologies afdeling.

Het volgende diagram is het organogram van heel Ordina. De afdeling waar ik werkzaam was, valt onder Technologie & Competenties.



Binnen Technologie & Competenties is het bedrijf op de volgende manier ingericht. De afdeling waar ik werkzaam ben, is J-Technologies, maar mijn opdracht heeft ook veel te maken met de afdeling SMART Technologies.



Binnen J-Technologies heb ik een Business Unit Manager en een begeleider. De opdrachtgever is van SMART Technologies.

3. De opdracht

De opdracht was om een spraakanalyselibrary voor presentaties te ontwikkelen.

3.1. Uitgangssituatie

De situatie bij aanvang van het afstuderen was als volgt.

Bij aanvang van het afstuderen was er nog geen bestaand project waarop moest worden voortgewerkt. Wel zal het resultaat van het project worden gekoppeld aan meerdere andere projecten die elk verantwoordelijk zijn voor een aspect van de analyse van de presentaties. Uiteindelijk worden al deze onderdelen aan elkaar gekoppeld en kan de gebruiker zonder hulp van een expert zijn presentatievaardigheden trainen.

Ik werd tijdens de opdracht begeleid door Frank Verbruggen. Ik heb hem elke week op een vast tijdstip gesproken. Deze gesprekken waren altijd prettig en hebben zeker bijgedragen aan de vormgeving en voortgang van de opdracht.

Ik heb gebruik gemaakt van de Eclipse IDE. Het Spring framework en LinkedData Virtuoso, zoals besproken in het afstudeerplan, bleken niet nuttig voor dit project.

3.2. Probleemstelling

Binnen de afdeling van Ordina die de omscholing en trainingen verzorgt, worden o.a. presentatietrainingen gegeven. Er is bij deze trainingen behoefte aan software die bepaalt hoe goed een presentatie is. Met zo'n analyse kan objectief feedback gegeven worden over hoe mensen beter hun stem kunnen gebruiken en uitspraak kunnen verbeteren.

3.3. Doel van de opdracht

Het doel van de opdracht was om een spraakanalyselibrary te ontwikkelen. Deze library moest op basis van alleen de audio van een presentatie feedback kunnen geven over de presentatie.

Om zo'n analyse te doen, moest er gebruik gemaakt worden van een spraakherkenningslibrary. Deze library heb ik gekozen aan de hand van een kort onderzoek aan het begin van het project. Op basis van de data die de spraakherkenningslibrary uit de audio haalt, wordt vervolgens een analyse uitgevoerd.

3.3.1. Wat is een presentatie?

Ik heb voor dit project, in overleg met mijn begeleider, de grens bepaald tussen wat wel en niet een presentatie is. Deze grens is vrij strikt, maar de software is uiteindelijk alleen bedoeld voor presentaties.

- Een informatieve presentatie is waar de software voor bedoeld is. Dit is uiteraard een presentatie.
- Een speech, zoals bijvoorbeeld een officiële mededeling van Obama, wordt wel gezien als presentatie.
- Een documentaire, vaak met achtergrondmuziek voor de sfeer, wordt niet gezien als presentatie.
- Muziek wordt niet gezien als een presentatie.

Met deze definitie in gedachten heb ik de spraakanalyses in elkaar gezet.

3.3.2. Potentieel probleem

Een potentieel probleem met de opdracht was bewijzen dat de resultaten van de analyse kloppen met de werkelijkheid. De resultaten van een analyse bestaan uit scores voor verschillende aspecten van een presentatie en vervolgens een algemene score.

3.3.3. Mogelijkheden en oplossing

De struikelblokken voor het bewijzen van de validiteit van de analyse waren de volgende:

- Bepalen wat de verwachte score is voor een presentatie
- Bepalen hoe de analyse zal worden verbeterd op basis van de daadwerkelijke score
- De kwaliteit van de gebruikte spraakherkenning

Om een betrouwbare analyse te krijgen, moeten de verbeteringen gebaseerd worden op veel data. Dit is een belangrijk punt voor de volgende redeneringen.

De verwachte score

Voor het bepalen van de verwachte score zijn er twee opties. Deze wordt bepaald door mensen of door een algoritme op basis van bestaande data. Een voorbeeld van dit soort data is de views, likes en dislikes op YouTube.

De verwachte score laten bepalen door mensen zou veel meer tijd kosten. Een mens kan een presentatie niet sneller beoordelen dan de duur van de presentatie. Ook zou, om persoonlijke voorkeuren voor een spreker of accent of iets dergelijks de data niet te laten beïnvloeden, hetzelfde bestand beoordeeld moeten worden door meerdere mensen.

Een beoordeling op basis van YouTube-data zou geautomatiseerd kunnen, tientallen keren sneller uitgevoerd kunnen worden en bovendien gebaseerd zijn op de mening van miljoenen mensen. Het enige nadeel hieraan is dat het onderwerp van een presentatie ook het aantal views, likes en dislikes kan beïnvloeden. Daarentegen is dat met beoordeling door mensen ook het geval. Echter moeten niet-presentatie dan wel herkend kunnen worden. Deze data is namelijk niet representatief voor het beoordelen van een presentatie, maar zal wel vergelijkbare views, likes en dislikes hebben. Daarmee is de data niet vergelijkbaar en moet deze gefilterd worden.

Ik heb ervoor gekozen de bestaande data van YouTube te gebruiken.

Verbeteren resultaat analyse

Het zou mij zeer verbaast hebben als de uitkomsten van de analyse niet afwijken van de verwachte scores. Er moest dus een oplossing gevonden worden voor het verbeteren van het resultaat dat de analyse geeft op basis van de verzamelde data van YouTube. Al snel kwamen mijn begeleider en opdrachtgever met het idee daarvoor machine learning te gebruiken. Zelf had ik hier geen ervaring mee, maar zag hier eerder een interessante uitdaging dan een probleem in.

Kwaliteit externe library

De kwaliteit van de gebruikte spraakherkenning is van grote invloed op de analyse. Op basis van de uitkomsten van het onderzoek is er besloten het mogelijk te maken meerdere spraakherkennings-API's te gebruiken. Op deze manier zou ook bepaald kunnen worden hoe goed een library is, door bijvoorbeeld de accuratesse van libraries met elkaar te vergelijken. Hiermee kan gezegd worden in hoeverre de kwaliteit van de externe library de resultaten beïnvloedt.

4. Beschrijving van de werkzaamheden

Hier zal ik per timebox beschrijven welke werkzaamheden ik heb uitgevoerd, wat de problemen waren, welke mogelijke oplossingen er waren, de motivatie achter mijn keuze en waar van toepassing de resultaten.

4.1. Timebox 1 – Plan van aanpak en onderzoek

In de eerste timebox heb ik het plan van aanpak gemaakt en heb ik een onderzoek uitgevoerd naar welke open-source libraries of web-API's geschikt zijn om te gebruiken voor de spraakherkenning.

4.1.1. Plan van aanpak

Het plan van aanpak is te vinden in bijlage A. De belangrijkste punten hieruit zijn de volgende.

Projectopbouw

Bij het project heb ik timeboxing, zoals gebruikelijk bij Scrum, toegepast. Ik heb hiervoor gekozen deels omdat Scrum de standaard is binnen Ordina. Daarnaast had ik wekelijks op vrijdag een afspraak met mijn begeleider. Door de timeboxing toe te passen, was ik er zeker van dat ik bij het gesprek de onderdelen waar ik aan werkte af of bijna af had. Zo kon ik altijd het onderdeel waar ik op dat moment aan werkte laten zien en/of bespreken. Ook zou ik mijn planning makkelijk aan kunnen passen zonder het ontwikkelen van een product halverwege te moeten onderbreken. Omdat ik zo'n korte timeboxes had vergeleken met de gebruikelijke 4 weken van Scrum, leek het mij tijdverspilling om aan het eind van elke timebox te reviewen.

Planning

Bij het maken van de planning heb ik alleen voor de eerste twee timeboxes vastgelegd wat ik zou doen. Dit is het volgende:

Timebox 1:

- Plan van aanpak
- Kort onderzoek open-source libraries

Timebox 2:

- Eerste UML ontwerp

De reden dat ik alleen de eerste twee timeboxes heb vastgelegd, is dat de verdere verloop van het project in grote mate afhankelijk was van het onderzoek naar open-source spraakherkenningslibraries.

De verdere planning was een overzicht van de werkzaamheden die ik in alle timeboxes samen wilde uitvoeren. Dit zijn de volgende:

- Opstellen, prioriteren en overeenstemmen requirements. Dit aan de hand van interviews/gesprekken met medewerkers van Ordina. De requirements worden ingedeeld aan de hand van het MoSCoW-principe.
- Ontwerp van het product. Dit zal gedaan worden m.b.v. UML in zowel analyse- als designdiagrammen
- Bouw van het product
- Refactoren van de code

- Testen. De library zal aan de hand van een testplan per module en vervolgens ook in zijn geheel getest worden.

Na het uitvoeren van deze werkzaamheden moest het product alleen nog overgedragen worden. Bij de overdracht hoorde ook het schrijven van de API-documentatie van de ontwikkelde library.

Oplevering

De volgende documenten en onderdelen zal ik opleveren:

- Plan van aanpak
- Onderzoeksrapport over de librarykeuze
- Ontwerp van het product
- Spraakanalyselibrary
- API-documentatie
- Proof-of-concept applicatie

4.1.2. Achterhalen requirements

Het achterhalen van requirements heb ik over het algemeen gedaan in gesprekken met Richard van Tilborg, de opdrachtgever. Ook kon ik bij mijn begeleider terecht als ik ergens over twijfelde.

Ik heb geen technieken gebruikt om requirements te achterhalen. Ik achtte dit niet nodig omdat de gesprekken over het algemeen kort en to-the-point waren. Het zou naar mijn idee tijdverspilling zijn. De heer van Tilborg liet een groot deel van de technische details aan mij over.

Ik was oorspronkelijk ook van plan gesprekken te voeren met de presentatoren van Ordina. Ik wilde dit doen om te achterhalen wat belangrijke aspecten zijn in een presentatie. Uiteindelijk heb ik hier van afgezien, omdat er al mogelijke aspecten naar voren kwamen in de gesprekken met Richard. Ook zou er veel tijd gaan zitten in het voorbereiden, houden en verwerken van deze interviews. Ik denk niet dat ik er voor het project zo veel voordeel van had gehad dat het de tijd waard was geweest.

Ik heb de requirements ingedeeld volgens het MoSCoW-principe. Ik heb het MoSCoW-principe toegepast om overzicht te creëren en zo nodig requirements te schrappen vanwege tijdgebrek.

4.1.3. Onderzoek open-source Java spraakherkenningslibraries

Ik heb om te beginnen onderzoek gedaan naar open-source Java spraakherkenningslibraries. Dit heb ik gedaan m.b.v. de longlist-shortlist methode. Zie Bijlage B voor het complete onderzoek.

Voor het onderzoek heb ik een kort gesprek gehad met Richard. Hierin heb ik zijn wensen wat betreft de te ontwikkelen library geprobeerd te achterhalen. Ik vond het voor zo'n kort gesprek niet nodig een techniek voor het achterhalen van requirements te gebruiken.

De requirements die in het gesprek met Richard naar voren kwamen zijn:

Requirement	Prioriteit
De library moet geschreven worden in Java	Must
De library moet Nederlands ondersteunen	Should
Verstaanbaarheid analyseren	Should
Snelheid analyseren	Could
Stiltes analyseren	Could
Aantal keer 'eh' tellen	Could
Verloop toonhoogte analyseren	Could

Op basis van deze requirements en een aantal logische eisen aan de spraakherkenningslibrary heb ik de eisen voor de longlist en shortlist opgesteld.

Longlist

De longlist bestond uit alle Java spraakherkenningslibraries die ik kon vinden op SourceForge, GitHub. Libraries die ik op andere websites vond, stonden uiteindelijk allemaal op één van beide.

Vervolgens keek ik bij elke library of API naar de volgende punten:

- Is de library te gebruiken in Java?
- Heeft de library de gewenste functionaliteit?
- Wordt er actief gewerkt aan de library?

Op basis van deze eisen ben ik tot de volgende tabel gekomen:

Library/API	Gebruik met Java	Gewenste functionaliteit	Onderhoud library
Srt-translator	V	V	V
Java Speech API	V	?	?
NoiseGator	V	X	
Voce	V	V	V
HMM Speech Recognition	X		
Jarvis	V	X	
Activey/VoiceRecognition	V	X	
Praat	X		

Shortlist

Al snel bleek dat elke van de libraries of API's op de longlist gebruik maakt van één of meerdere van de volgende API's: Sphinx4, Google Speech API en Java Speech API. De libraries die aan de eisen van de longlist voldeden waren: Srt-translator en Voce. Deze twee en de onderliggende API's vormden de shortlist.

Bij de shortlist heb ik gekeken naar de volgende punten:

- Onderliggende API
- Ondersteuning Nederlands
- Ontwikkelfase van de library

Met deze punten kwam ik tot de volgende tabel:

Library/API	Onderliggende API	Functionaliteit	Ondersteuning Nederlands	Ontwikkelfase API
Voce	Java Speech API en Sphinx4	Beperkt	X	Released
Srt-translator	Google Speech API	Beperkt	V	Beta
Sphinx4	-	Uitgebreid	X	Released
Java Speech API	-	?*	?*	Released

* De Java Speech API compileerde niet meteen na download. Omdat het er veelbelovend uit zag, besloot ik één dag te besteden aan het aan de praat krijgen van deze library. Dit is niet gelukt.

Hierbij weegt de functionaliteit zwaarder dan de andere punten. Dit omdat er met beperkte functionaliteit niet veel gezegd kan worden over de kwaliteit van de presentatie. Bijvoorbeeld bij de Google Speech API ondersteunt Nederlands wel, maar geeft alleen een transcriptie en mate van zekerheid. Hier is niet een hele analyse op te baseren.

Conclusie

De conclusie van het onderzoek is:

De functionaliteit van Voce is matig ten opzichte van Sphinx. Ik zie geen reden om een beperktere API te gebruiken als de onderliggende API ook beschikbaar is.

De Srt-translator is de enige API die Nederlands ondersteunt. Echter is de functionaliteit zeer beperkt en is het project nog in de bèta-fase. Daarnaast is er de kans dat Google besluit de API te sluiten.

De Java Speech API is een mindere optie omdat de kwaliteit van de spraakherkenning niet zeker is en het Nederlandse akoestische model nog getraind moet worden. Dit is mij te veel onzekerheid om een project op te baseren.

Sphinx4 is een uitgebreid qua functionaliteit en komt het met getrainde akoestische modellen. Sphinx4 laat dan wel te wensen over in de API documentatie, maar er de tutorials zijn zeer duidelijk. Er zit veel voorbeeldcode bij de tutorials. Ook is het een veel gebruikte library waardoor er veel problemen al opgelost zijn door anderen.

Uit deze en voorgaande punten concludeer ik dat Sphinx4 de beste keuze is.

Probleem

Zoals in bovenstaande conclusie is Sphinx4 library de beste optie. Echter ondersteunt deze geen Nederlands ondersteunt. Dit is wel een vrij grote wens van de opdrachtgever. Sphinx Nederlands aanleren zou een lange termijn project worden.

Mogelijke oplossingen

Aangezien Sphinx4 niet voldoet aan één van de eisen van de opdrachtgever, was ik niet zeker over mijn keuze. Omdat de definitieve keuze voor een library een vrij grote beslissing is, heb ik deze niet in mijn eentje genomen, maar het probleem aan mijn begeleider en opdrachtgever voorgelegd.

Gekozen aanpak

Ik heb dit probleem met mijn begeleider en mijn opdrachtgever besproken. Een lange termijn project opstarten om Sphinx Nederlands aan te leren was voor mijn begeleider geen optie. Op dat moment kwam toch de Google Speech API weer ter sprake. Samen kwamen we tot een oplossing die ik niet had verwacht. Dit heeft de opdracht ietwat complexer gemaakt. De gekozen oplossing wordt beschreven onder de volgende vier kopjes:

Standaardisatie API

Voor nu is de Google Speech API goed genoeg om Nederlands mee te analyseren. Er zal in de toekomst waarschijnlijk een andere API of library gebruikt moeten worden om Nederlands te analyseren. Deze nieuwe library moet met niet al te veel moeite aan het systeem kunnen worden toegevoegd. Daarom zal ik een gestandaardiseerde API definiëren die een brug zal vormen tussen de spraakanalyse en welke API of API's er dan ook achter komen. Het bouwen van een gestandaardiseerde API biedt ook mogelijkheden. De verschillende API's zouden bijvoorbeeld ook met elkaar kunnen worden vergeleken.

Meerdere API's

Door een gestandaardiseerde API te definiëren is het mogelijk meerdere API's tegelijk te gebruiken.

Om Nederlands te kunnen analyseren besloten we de Google Speech API te gebruiken. De Google Speech API was volgens het onderzoek daar de beste kandidaat voor. Voor de analyse van Engels kan zowel de Google Speech API als Sphinx4 gebruikt worden.

Het plan was dat ik met deze libraries aan de hand van een tiental audiobestanden een algoritme in elkaar zet. Dit algoritme kan op basis van de output van de API's de audio beoordelen. Dit algoritme zal moeten werken op basis van een aantal parameters, zoals bijvoorbeeld hoe zwaar de verstaanbaarheid moet meewegen in het uiteindelijke cijfer. Een tiental bestanden is in dit geval genoeg omdat slechts een benadering van de uiteindelijke parameters voldoende was. De parameters zullen aangepast worden aan de hand van de eerder besproken bestaande data, waarschijnlijk van YouTube m.b.v. machine learning.

Machine learning

Het plan was om de resultaten van het algoritme te verbeteren aan de hand van bestaande data. De data van YouTube is via de API vrij toegankelijk. Op basis van de views, likes en dislikes kan m.b.v. machine learning het algoritme steeds beter gemaakt worden.

Op deze manier kan ook met de uitkomsten van het resulterende algoritme bepaald worden hoe goed een API is. Er kan bijvoorbeeld gekeken worden naar in hoeverre de beoordeling van de API

afwijkt van de beoordeling opgemaakt uit de data van YouTube, oftewel de accuratesse van de API. Deze kan vervolgens vergeleken worden met andere API's.

De machine learning kan worden toegepast op meerdere API's, maar voor dit project heb ik het bij één library gehouden. Wel zullen sommige parameters verschillen per API. Hier heb ik dus voor toekomstige API's rekening mee gehouden.

4.2. Timebox 2 – Ontwerp en aanpassing werkwijze

In de tweede timebox heb ik het eerste UML ontwerp gemaakt. Ook heb ik een gesprek gehad met mijn begeleider en daarin de werkwijze bepaald.

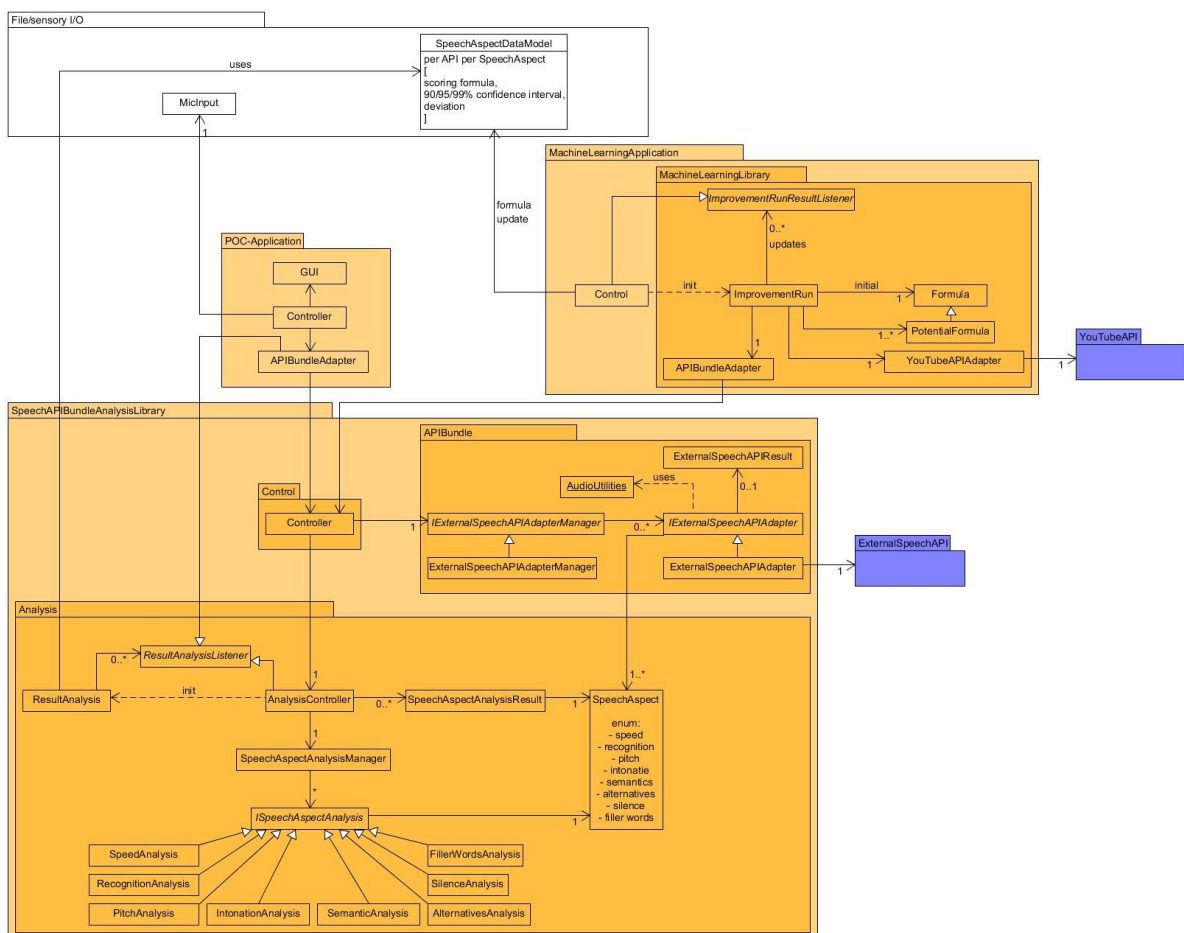
4.2.1. UML ontwerp

Ik heb UML gebruikt voor het ontwerp van de te ontwikkelen library. Ik heb alleen classdiagrammen gemaakt. In deze classdiagrammen heb ik de structuur van de library weergegeven. Als dat het diagram duidelijker maakte, heb ik ook interactie van classes met andere libraries of bestanden/hardware weergegeven. Naast de structuur van de library heb ik ook de relaties tussen de verschillende classes weergegeven. Het weergeven van alle variabelen en methoden van alle classes vond ik niet nodig. Hier was veel tijd in gaan zitten, terwijl het even later alweer veranderd was. Deze beslissing heeft mij veel tijd bespaard en ik heb er geen hinder van ondervonden.

Ik heb over de loop van het project het UML ontwerp meerdere keren geüpdatet. Bij zo'n update heb ik het nieuwe ontwerp niet alleen in dit document gezet, maar ook in één van de bijlagen. Zo staan de veranderingen overzichtelijk bij elkaar. Ik heb ook bij elke verandering uitgelegd waarom ik de verandering heb gedaan.

Eerste ontwerp

Het eerste ontwerp was het volgende diagram:



Dit ontwerp was een vrij uitgebreid diagram. Mijn begeleider vond dit eerste ontwerp veel te gedetailleerd. Ik had beter al wat kunnen doen aan de implementatie van een aantal classes. Op deze manier neigde ik meer naar een waterval-achtige aanpak. We hebben afgesproken dat ik vaker kleinere stappen ga maken, i.p.v. in één keer alles goed te willen doen. Ik ben echter wel blij dat ik er in het begin wat meer tijd in heb gestoken. Zo heb ik meteen over het grotere geheel nagedacht en me al wat kunnen verdiepen in de verschillende onderdelen van de te ontwikkelen library.

Alle oranje onderdelen heb ik zelf geïmplementeerd. De blauwe onderdelen zijn bestaande onderdelen.

4.2.2. Werkwijze

De uiteindelijke aanpak week wat af van hoe ik het heb beschreven in het afstudeerplan. Het is volgens mijn begeleider, en daar ben ik het mee eens, beter om 'herkenbaar te falen'. Dit houdt in dat ik vaker iets oplever. Het hoeft dan niet perfect te zijn, maar het moet fout zijn op zo'n manier dat aangewezen kan worden waar de fouten zitten. Ik zal wel in timeboxes blijven werken. Voor de planning van deze timeboxes heb ik een backlog bijgehouden. Hierin heb ik in grote lijnen bijgehouden wat er moest gebeuren. Aan het begin van elke week deelde ik de eerstvolgende punten op in kleinere blokken voor die week. Deze planningen staan aan het begin van elke timebox.

Na het maken van deze afspraak heb ik vaker tussenproducten opgeleverd en ben ik ook concreter gaan werken.

4.3. Timebox 3 – Eerste implementaties API's en aspecten

4.3.1. Planning

In timebox 3 stond het volgende op de planning:

- Testbestanden van een goede en een slechte presentatie of speech zoeken en deze klaar maken voor gebruik
- Een adapter schrijven voor Sphinx4 en Google Speech API
- Beide adapters kunnen gebruiken op basis van een testbestand
- Handmatig schrijven of op internet vinden van een transcript voor het gekozen testbestand
- Aan de hand van deze transcripten bepalen hoe accuraat Sphinx4 en Google Speech zijn
- Zoeken naar aspect dat zeer verschillend is bij Obama en Palin
- Implementeren aspect
 - o Implementatieplan
 - o Implementeren en testen aan de hand van Obama en Palin
- Detecteren non-presentatie
 - o Implementatieplan

4.3.2. Testbestanden

Voor het testen heb ik op YouTube gezocht naar een speech van Obama en één van Sarah Palin. Ik heb YouTube gekozen omdat dit ook de uiteindelijke bron van data wordt. Obama leek mij een goede optie omdat hij de president van de Verenigde Staten is en dus veel ervaring heeft. Ik heb gekozen voor de kortste video van Obama die er was, om een test zo kort mogelijk te houden zonder een speech te moeten afbreken. Ik heb voor Sarah Palin gekozen omdat zij erom bekend staat relatief slechte speeches te geven. Ze is echter wel een ervaren spreker en zal dus geen compleet belabberde speeches geven.

4.3.3. Implementatie API's

Hier zal ik per API kort omschrijven wat ik heb gedaan om de eerste opzet voor elke API te maken. Details van de implementatie zijn te vinden in het ontwerp bestand.

Sphinx4

Sphinx4 is een library in Java, die toegevoegd moet worden aan het project. Voor Sphinx moet het audiobestand de volgende eigenschappen hebben:

- Samplerate: 16kHz
- Samplesize: 16kB
- Mono/stereo: Mono
- Format: .wav

Het bestand van Obama heb ik m.b.v. Audacity aangepast naar deze eisen. Zoals in het eerste ontwerp ook al aangegeven, zal er waarschijnlijk een class komen die deze conversies kan afhandelen. Dit is geen directe eis van de opdrachtgever geweest, maar naar mijn idee een logische stap.

Voor het opzetten van de eerste test voor Sphinx heb ik de tutorials op de website gevolgd. Hier staat vrijwel alles in om een simpele eerste versie te maken. Na refactoren van de code bestaat de adapter uit de volgende classes:

- SphinxRecognizer. Deze wordt aangeroepen vanaf buiten de adapter. Er wordt een standaard configuratie ingeladen, die voor herkenning van Engels. Vervolgens wordt de SphinxSpeechAdapter, een Iterable, gebruikt om de resultaten te printen of naar een file te schrijven of te printen.
- SphinxSpeechAdapter. Dit is een Iterable, aangemaakt door de SphinxRecognizer. Er kan geïtereerd worden over dit object tot er geen volgend resultaat meer is.

Google Speech API

De Google Speech API is een web-API. Deze moet worden aangeroepen met een http-request. Het antwoord van de API is een json-response met daarin een aantal hypotheses voor een transcriptie. Bij de eerste wordt ook een 'confidence' gegeven. De confidence is een indicatie van hoe zeker de API is over de transcriptie. Er wordt geen confidence gegeven over de alternatieven. De API accepteert alleen bestanden van maximaal 15 in het .flac-formaat verwerken.

Voor de Google Speech API heb ik de code van de Srt-translator aangepast. Dit was één class met alles in één methode. Hier heb ik een class van gemaakt die enigszins lijkt op de SphinxRecognizer.

In de oorspronkelijke class werd al gebruik gemaakt van een library die een audiobestand kan opsplitsen en omzetten naar .flac-bestanden van 15 seconden. Voor deze library heb ik een aparte adapter gebruikt. Deze heb ik later gebruikt voor de conversie van alle bestanden.

Uiteindelijk bestaat de adapter voor Google uit de volgende classes:

- GoogleSpeechRecognizer. Deze wordt aangeroepen vanaf buiten de adapter. Hier wordt de GoogleSpeechAdapter aangeroepen en de resultaten daarvan verwerkt. De resultaten kunnen geprint of naar een bestand worden geschreven.
- GoogleSpeechAdapter. Een Iterable<InputStream>. Hier wordt de Google Speech API aangeroepen.
- AudioRecoder. Deze class maakt gebruik van een externe library om audio te kunnen aanpassen naar het gewenste formaat.

In eerste instantie werd het opvragen van de ResultStream van de connectie nog gedaan in de GoogleSpeechRecognizer. Na refactoring is de GoogleSpeechAdapter een Iterable<InputStream> geworden en hoeft de GoogleSpeechRecognizer alleen de inhoud van deze stream te verwerken.

4.3.4. Standaardisatie van de output

De standaardisatie van de resultaten zal uitgebreid worden wanneer dat nodig is voor de implementatie van een nieuw aspect. Dit om onnodig werk te voorkomen en daarmee tijd te besparen. Het zal worden aangegeven als een aanpassing invloed heeft op de implementatie van de standaardisatie van de output.

4.3.5. Transcripties

De transcripties staan in Bijlage D. De belangrijke resultaten hieruit zijn de precisie van de API's voor beide:

- Obama met Sphinx4: 81%
- Obama met Google: 85% met ontbrekende resultaten.
- Palin met Sphinx4: 64%
- Palin met Google: Compleet ontbrekende resultaten.

Deze nauwkeurigheid heb ik laten berekenen door een online tool:

http://www.tools4noobs.com/online_tools/string_similarity/.

Lege responses

Wat als eerste opviel is dat er bij de Google Speech API resultaten ontbraken. Deze kwamen terug als lege json-responses. Ik heb geprobeerd de ogenschijnlijke limiet die er op de requests zit te omzeilen door vertragingen in te bouwen. Echter is het voorgekomen dat het eerste request van de dag compleet leeg terug kwam. Het lijkt willekeurig te zijn wanneer er wel en niet een transcriptie in de response zit. Hierom acht ik de Google Speech API onbetrouwbaar.

Ik ben vanaf dit punt verder gegaan op basis van alleen de Sphinx4 library.

Opmerkingen

Vergeleken met de 81% van Obama, is 64% een stuk slechter. Dit betekent dat er bijna twee keer zo veel woorden verkeerd zijn verstaan. Sarah Palin is blijkbaar een stuk lastiger te verstaan. Dit viel mij al op bij het maken van de handmatige transcriptie. Ik had vrij vaak moeite haar te verstaan. Ik heb meerdere keren een zin 4 of 5 keer moeten beluisteren.

Wat ook opviel bij het maken van de transcriptie, is dat het aantal keer dat Sarah Palin 'ehm' zegt slechts 3 is. Wel herhaalt ze woorden, dit is misschien haar manier om stiltes te vullen i.p.v. 'ehm' te zeggen. Ze herhaalt een woord soms wel 4 of 5 keer.

4.3.6. Aspecten

In plaats van op verstaanbaarheid en articulatie te focussen, heb ik besloten eerst aspecten te implementeren die minder complex zijn. Er zal gezocht worden naar een aspect waarin Obama en Palin zeer verschillend zijn. Op basis van dit aspect kan vervolgens een analyse worden geïmplementeerd. Zoals te zien in het eerste UML ontwerp, heb ik al wat ideeën voor mogelijk bruikbare aspecten: Snelheid, toonhoogte, intonatie, grammatica of andere semantische aspecten, alternatieven, stiltes en tijdrek-woorden.

Confidence

Het meest voor de hand liggende aspect van de transcriptie is hoe zeker de API is van de transcriptie, de confidence. Het is in Sphinx mogelijk om uit het SpeechResult dat de spraakherkenning geeft, de verstande woorden te halen. Van deze woorden kan de confidence worden opgevraagd. Dit is een getal tussen de 0 en 1 dat de verwachte accuratesse weer zou moeten geven. Echter bleek al bij de eerste test het confidence niveau altijd 1 te zijn, ook als de herkende woorden niet juist zijn. Hieronder de output voor de eerste woorden in de speech van Obama. Obama begint zijn speech met "Good evening.". Toch wordt er met een confidence van 1 "good enough" verstaan.

```
Result: <s> good enough </s>
{good, 1,000000, [500:630]}
{enough, 1,000000, [630:870]}
{</s>, 1,000000, [870:1470]}
```

De confidence van de woorden is dus niet bruikbaar. Wat hier wel bruikbaar aan is, is de laatste regel, die de stilte tussen 0.870 en 1.470 seconden representeert. Hiermee heb ik later de analyse van de stiltes op gezet.

Alternatieven

Om toch de verstaanbaarheid te kunnen bepalen, heb ik vervolgens naar de alternatieven voor de verstande woorden gekeken. Elke transcriptie heeft een aantal alternatieve transcripties.

Relatie aantal woorden – aantal alternatieven

Om het aantal alternatieven te gebruiken om te bepalen hoe goed een presentatie is, moet ik weten hoe deze data er uit ziet. Is er bijvoorbeeld een relatie tussen het aantal woorden in een fragment en het aantal alternatieven?

Bij het opvragen van de 1.000.000 beste alternatieven voor hetzelfde fragment, "Good evening" van Obama, kwam de volgende lijst alternatieven er uit:

- <s> couldn't </s>
- <s> good enough </s>
- <s> and in a </s>
- <s> but in </s>
- <s> put in a </s>
- <s> when </s>
- <s> in a </s>
- <s> good enough i </s>
- <s> in </s>
- <s> women </s>
- <s> it </s>
- <s> within </s>
- <s> within a </s>

- <s> good in a </s>
- <s> good in </s>
- <s> when i </s>
- <s> but in a </s>
- <s> when a </s>
- <s> could it </s>
- <s> didn't </s>
- <s> when it </s>
- <s> good </s>

Hieruit is op te maken dat de spraakherkenning van Sphinx de juiste woorden niet eens als alternatief geeft. Dit kan komen doordat de gesproken woorden niet in het woordenboek van Sphinx zitten, maar dat lijkt mij sterk voor “Good evening”.

Daarnaast bestaat deze lijst uit 22 opties, terwijl ik om 1.000.000 vroeg. Het vermoeden is dat als deze functie er meer had teruggegeven, daaruit geconcludeerd kan worden dat de spraakherkenning meer moeite had het te verstaan. Echter is het aantal alternatieven ook afhankelijk van het aantal woorden in het audiodeel. Dit is te zien in het volgende resultaat:

Gesproken tekst: Tonight, I can report to the American people and to the world

Resultaat: <s> <sil> tonight <sil> [speech] <sil> i can report to the
american people <sil> and to the world </s>

Sphinx geeft hier 798 alternatieven voor. Deze laat ik weg omdat het niet nuttig zou zijn en meerdere pagina's zou vullen.

Aangezien Sphinx elke combinatie van alternatieve woorden ook terug geeft, zal het aantal alternatieven een exponentiële functie van het aantal woorden zijn.

Relatie aantal alternatieven – kwaliteit presentatie

Nu ik een eerste indruk heb van de alternatieven die worden gegeven, kan ik dit toepassen op de hele testbestanden.

Ik heb een lijst gegenereerd met daarin het aantal verstande woorden en het bijbehorende aantal alternatieven van elk fragment. Op basis van het vermoeden van de relatie tussen het aantal woorden en het aantal alternatieven heb ik categorieën gemaakt op basis van het aantal woorden. Ik heb dit gedaan voor zowel Palin als voor Obama.

De resultaten hiervan heb ik in een Excel-bestand gezet. Vervolgens heb ik van het geheel en van elke categorie het gemiddelde en de standaarddeviatie (SD) laten berekenen door gebruik te maken van de formules in Excel. Ook heb ik de resultaten van Obama met die van Palin vergeleken door deze door elkaar te delen.

Ik heb voor de standaarddeviatie gekozen omdat ik benieuwd was naar hoe verspreid de data was. Daarnaast kun je naar mijn idee wennen aan een slechtere uitspraak of een accent. Het is lastiger als iemand steeds verandert qua uitspraak.

Het resultaat hiervan staat in de volgende tabel:

			0-10		11-20		21+	
	Gem.	SD	gem.	0-10 SD	gem.	11-20 SD	gem.	21+ SD
Obama	16834,07	89487,5	552,6538	1775,494	46660,84	91541,89	1000000	
Palin	46982,79	196167,9	686,8632	1490,747	129485,1	293571,3	1000000	0
Obama/Palin	0,358303	0,456178	0,804605	1,19101	0,360357	0,311822	1	

In deze tabel is te zien dat er bij Obama in alle categorieën gemiddeld minder alternatieven zijn voor de verstane tekst. Over het algemeen is de standaarddeviatie bij Obama ook kleiner.

Hoewel dit uiteraard een kleine test is, wekt dit wel het vermoeden dat, naast een kleiner aantal alternatieven, ook een kleinere standaarddeviatie betekent dat de verstaanbaarheid beter is. Mijn gedachte achter het voordeel van een kleinere standaarddeviatie is dat mensen kunnen wennen aan een slechtere uitspraak. Als deze uitspraak over de hele presentatie genomen even slecht blijft, is dat makkelijker om te verstaan dan als het wisselt.

Wat interessant is aan deze tabel is vooral het verschil in de 11 t/m 20 woorden categorie. Deze extreme verschillen komen hoofdzakelijk doordat Palin in deze categorie fragmenten heeft met meer dan 1.000.000 alternatieven. Dit beïnvloedt de scores enorm. Bij Obama is dit niet het geval.

4.3.7. Implementatie alternatievenanalyse

Hier zal besproken worden hoe ik de alternatievenanalyse heb geïmplementeerd.

Aanpassingen standaardisatie resultaten

Voor de implementatie van het analyseren van de alternatieven heb ik in eerste instantie alleen het aantal woorden en het aantal alternatieven dat daarbij hoort gebruikt. Hiervoor zijn de volgende aanpassingen gedaan aan de standaardisatie van het resultaat:

- `StandardizedSpeechResult`. Een class met daarin een string van het resultaat en een `ArrayList` van `SpeechResultAlternatives`
- `SpeechResultAlternatives`. Een class met het aantal woorden en aantal alternatieven

Ook aangepast is de `SphinxSpeechAdapter`, zodat deze een `StandardizedSpeechResult` met `SpeechResultAlternatives` produceert en deze door de analyse(s) haalt. Deze analyse(s) zijn gedefinieerd in een `ArrayList` van `ISpeechAspectAnalysis`. De analyse heb ik als volgt in elkaar gezet:

- `ISpeechAspectAnalysis` interface met één methode:
`public void analyse(StandardizedSpeechResult);`
- `AlternativesAnalysis` is een implementatie van de `ISpeechAspectAnalysis`. Hier gebeurt de daadwerkelijke analyse. Er wordt een object van deze class aangemaakt in de `SphinxSpeechAdapter` voor in de lijst met `ISpeechAspectAnalysis` objecten.

Aan het eind van de spraakherkenning door de `SphinxSpeechAdapter` wordt voor elke `ISpeechAspectAnalysis` de `analyse()`-methode aangeroepen.

Analyse

`AlternativesAnalysis` analyseert het resultaat en vergelijkt het met de resultaten van Obama en Palin. Vervolgens worden statistieken van de resultaten en die van de vergelijking met Obama en Palin geprint. Op het moment is er nog geen optie om de resultaten terug te geven. Dit is uiteindelijk wel de bedoeling.

De analyse wordt gedaan op alle resultaten, maar ook op delen ervan, opgesplitst op het aantal verane woorden. Dit is gedaan omdat het aantal alternatieven drastisch oploopt naarmate het meer woorden worden. Er kan dan een analyse worden uitgevoerd op resultaten die op elkaar zouden moeten lijken, maar ook op het geheel.

De resultaten zijn opgedeeld in de categorieën Low (1 t/m 10 woorden), Med (11 t/m 20 woorden), High (21+ woorden) en Total (alle data). Vervolgens wordt er per categorie het gemiddelde, Avg, en de standaardafwijking, SD, weergegeven van het aantal alternatieven. Ten slotte worden deze gegevens vergeleken met de data over de presentaties van Obama en Palin en wordt aangegeven bij wie de presentatie het dichtst bij ligt. Het bepalen welke van de twee dichterbij ligt, wordt nu nog gedaan enkel op basis van het gemiddelde. Uiteindelijk moet ook de standaardafwijking worden meegenomen in dit oordeel.

Test

Als korte test heb ik een tweede presentatie van Obama geanalyseerd. Ik verwachtte uiteraard dat hier uit kwam dat deze presentatie bij Obama in de buurt ligt. Dit was ook het geval. Sterker nog, volgens de analyse is de tweede presentatie beter dan de eerste.

4.3.8. Niet-presentaties

Het doel hiervan is te kunnen bepalen of de ingevoerde audio wel van een presentatie is. Dit om te voorkomen dat er hele analyses worden uitgevoerd op audio die niet geschikt is.

Ik heb geprobeerd een audiobestand dat niet van een presentatie is aan de spraakherkenning te geven. Ik heb hiervoor de volgende drie bestanden gebruikt:

- Linkin Park – Papercut, de woorden hiervan zijn niet op normaal spraaktempo, maar een stuk langzamer. Dit, samen met de muziek die er achter zit, maakt het waarschijnlijk onverstaanbaar voor de library.
- Eminem – Lose Yourself, de spraak hiervan is wel op normaal tempo, maar er zit nog steeds muziek achter. Ik verwacht hier minder problemen dan bij Linkin Park, maar nog steeds dat het transcript niet in de buurt komt.
- Linkin Park – Numb (Acapella), dit is een opname van alleen de zang van een nummer. Er is dus geen muziek op de achtergrond. Echter blijft het muziek en geen presentatie.

Resultaten

Het resultaat van deze tests was zoals verwacht: de spraakherkenning kwam er niet uit.

- Bij Linkin Park – Papercut, werd niet een woord goed herkend. Er kwam hier en daar wel een paar woorden uit, maar de resultaten sloegen eigenlijk nergens op. Na enige tijd was er een OutOfMemoryError.
- Bij Eminem – Lose Yourself werden enkele woorden goed herkend. Maar, net als bij Linkin Park – Papercut, kwamen er nauwelijks resultaten uit en was er een OutOfMemoryError.
- Linkin Park – Numb (Acapella) kwam wel zonder OutOfMemoryError door de spraakherkenning heen, maar de resultaten waren veel minder dan in de audio en klopten niet met de tekst. Ook weken de resultaten extreem af van beide ijkpunten. Bijvoorbeeld een afwijking van 8800, terwijl het verschil tussen Obama en Palin 300 is.

Wat verder opviel, was dat er heel weinig resultaten uit de spraakherkenning kwamen. Denk hierbij aan 10 resultaten in 5 minuten, een resultaat per 30 seconden audio, tegenover een resultaat per 3.3 seconden audio voor Obama.

Ik heb in deze timebox nog geen echte check of iets een presentatie is of niet ingebouwd. Dit heeft niet echt haast en heb ik daarom voor een latere timebox gelaten.

4.3.9. Voortschrijdend inzicht aanpak

Er is een aantal punten uit mijn plan van aanpak dat ik achteraf anders had gepland:

Aan het eind van elke timebox het hele product testen is naar mij idee niet nodig. Aangezien de opdrachtgever en begeleider niet elke timebox een update hoeven over mijn voortgang, is er geen behoefte aan complete testrapporten. Ik zou deze alleen voor mijzelf maken. Zeker in de beginfase,

waarbij het product nog zo klein was dat een debug-run een complete test was, vind ik het maken van een testplanning en testrapport geen goede tijdsbesteding.

Elke timebox zou ik afsluiten met een review van het product. Dit is achteraf niet realistisch. Een review zou moeten worden uitgevoerd met meerdere mensen om alle fouten waar ik overheen kijk er uit te halen. Ik heb wel wekelijks een bespreking gehad met mijn begeleider, maar een complete review is iets te veel van het goede.

Aan eind elke timebox zou ik een demo geven. Dit is uiteraard ook zeer afhankelijk van anderen. Deze demo's stonden wel op de planning in de eerste weken, maar over het algemeen kwamen er niet veel mensen naartoe. Deze ene week was er niemand, of één iemand waar ik dan even mijn voortgang mee doorsprak. Dit is waarschijnlijk omdat het in de vakantieperiode was en er niet veel afstudeerders waren. Ook had ik in het begin van het project nog niet echt iets om te laten zien.

4.4. Timebox 4 – Stilte- en snelheidsanalyse

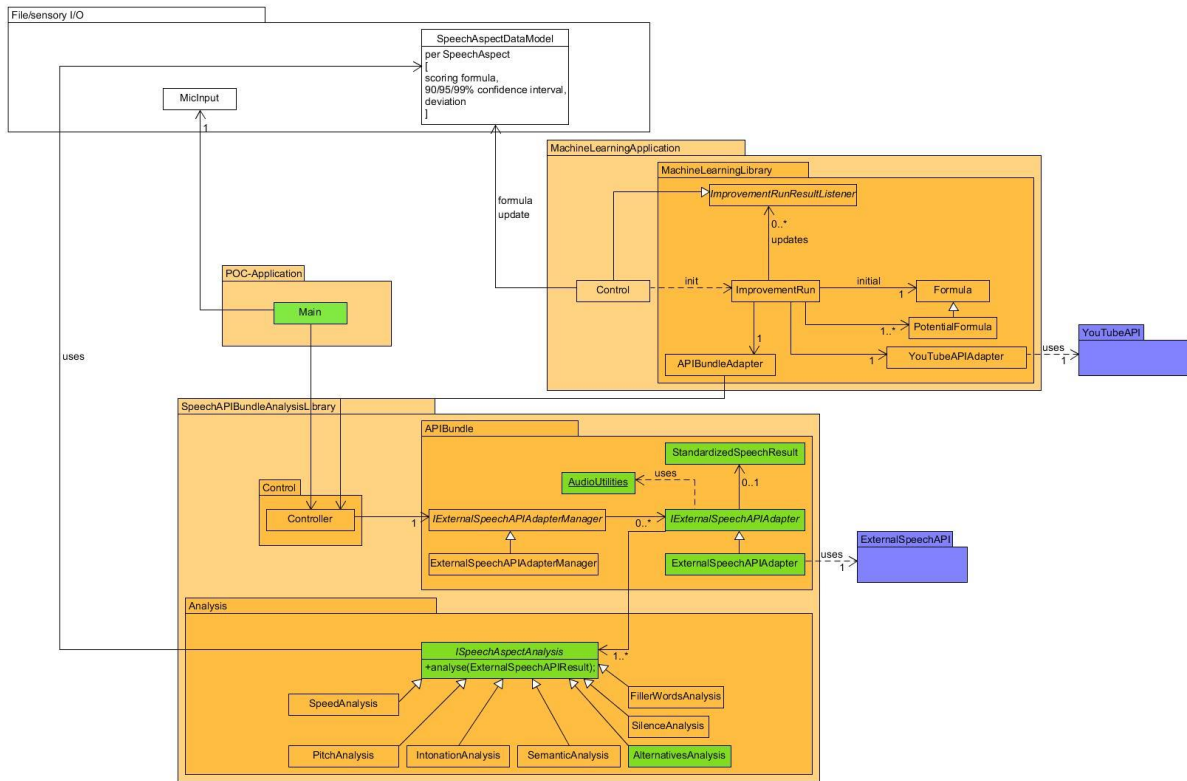
4.4.1. Planning

Voor timebox 4 stond het volgende op de planning:

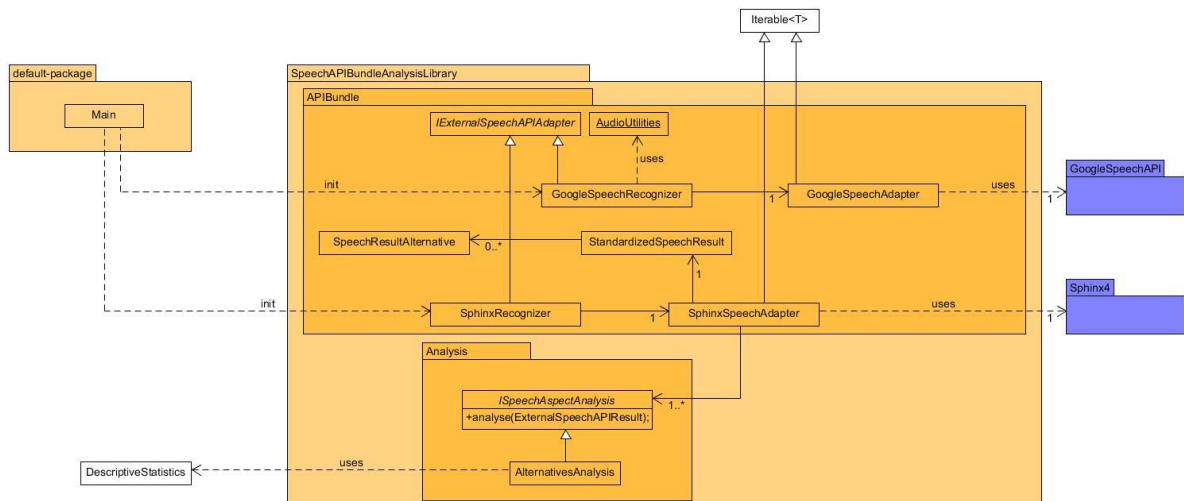
- Ontwerp document updaten. Veel beslissingen zijn nu niet gedocumenteerd, of alleen in afstudeerverslag.
- Realtime uitvoeren AlternativesAnalysis
- Refactoren
- Tweede analyse resultaat
 - o Zoeken naar aspect
 - o Implementeren aspect analyse
- Documenteren
- Meer audio bestanden zoeken. Minstens 2 per categorie.
- Testen met gevonden bestanden
 - o Testplan schrijven
 - o Tests uitvoeren
 - o Documenteren
- Versnellen
 - o Problemen in analyse
 - o Problemen in Sphinx
- Becijferen
 - o Eerste implementatie
 - o Finetunen
- GUI
 - o Textbox + Buttons
- YouTube-API
 - o Key e.d.
 - o Enkele video van YT halen
 - o Audio extracten
 - o Implementeren AudioUtilities, adapter voor library of eigen implementatie
 - o Audio omzetten en analyseren

4.4.2. Document updates

Bij deze eerste implementatie, zoals beschreven bij de vorige timebox, heb ik één aspect geïmplementeerd voor beide API's. De groengekleurde onderdelen zijn geïmplementeerd. Echter nog niet in de bedoelde structuur. De blauwgekleurde onderdelen zijn externe libraries die ik niet zelf implementeer. Alle oranje onderdelen moesten nog geïmplementeerd worden.



Het diagram van wat er tot nu toe is geïmplementeerd ziet er als volgt uit:



Beide recognizers hebben een Iterable<T> die de resultaten van de API ophaalt, maar die is niet een harde eis. Ik kan er naar mijn idee niet van uit gaan dat bij elke API een Iterable<T> kan worden toegepast.

4.4.3. Tussentijds uitvoeren analyse

Om de analyse uit te voeren tijdens de presentatie, heb ik een methode toegevoegd aan de interface `ISpeechAspectAnalysis`. Deze methode is `analyseProgress(StandardizedSpeechResult)`. De rede voor het onderscheid tussen de analyse tijdens en de analyse aan het eind van de presentatie is dat mensen tijdens een presentatie willen kunnen zien hoe goed ze het op dat moment doen. Wellicht kunnen ze dan aanpassingen maken in hun presentatiestijl. De aanpassing moet dan wel zichtbaar zijn in de tussenresultaten en niet verdwijnen in één groot gemiddelde. In `analyseProgress` kan bijvoorbeeld alleen rekening worden gehouden met de 20 laatste resultaten.

Dat mensen tijdens het presenteren willen zien hoe ze het doen, heb ik opgemaakt uit het volgende. Eind mei ben ik bij een andere afstudeerder langsgegaan, Hedwig Doets. Zij werkte aan de VR-kant van de uiteindelijke software. In de VR sta je met een PowerPoint voor publiek. Dit publiek kan reageren op hoe goed je presenteert door wel of niet afgeleid te raken. Ook is er een aantal grafieken met statistieken over de presentatie te zien. Mijn software o.a. moet de input voor deze grafieken gaan leveren. Dit heb ik gerealiseerd in de laatste timebox.

4.4.4. Implementatie aspecten

Als toevoeging op de alternatieven-analyse ga ik nog een aantal aspecten implementeren. Ik zal van verschillende aspecten achterhalen of deze aspecten bruikbaar zijn om onderscheid te maken tussen een goede en een slechte presentatie.

Stiltes

Een aspect van een presentatie dat vaak besproken wordt bij trainingen. Er moeten niet te veel, maar ook niet te weinig stiltes worden gelaten. Er is dus waarschijnlijk een ideale waarde of verhouding in één van de statistieken over de stiltes.

Aanpassingen standaardisatie resultaat

Voor het implementeren van de stiltes heb ik het volgende aan de StandardizedSpeechResult toegevoegd:

- SpeechResultSilence. Een class waarin het begin en einde van een stilte wordt bijgehouden.
- SpeechResultSpeech. Een class waarin de transcriptie en het begin en einde van spraak wordt bijgehouden.

Eerste analyse

Ik heb van de stiltes het volgende in de analyse opgenomen:

- Duur van de stiltes
- Duur van de spraak
- Standaarddeviatie van de duur van de stiltes
- Standaarddeviatie van de duur van de spraak
- Gemiddelde ratio spraak/stilte
- Standaarddeviatie van de ratio spraak/stilte

Ik heb de stilteanalyse geïmplementeerd en de twee eerder gebruikte presentaties van Obama en Palin ermee geanalyseerd. De resultaten zijn als volgt:

	Duur stiltes	SD stiltes	Duur spraak	SD spraak	Gem. ratio	SD ratio
Obama	590.48	472.09	2066.88	1362.20	0.44	0.59
Palin	426.11	634.49	1761.98	1677.86	0.44	1.02

Opmerkingen

Wat als eerste opviel is dat de gemiddelde ratio op het eerste gezicht hetzelfde is: 0.44. Het verschil was in werkelijkheid 0.0009. Wat wel verschillend is, is de duur van de stiltes en spraak. Obama laat langere stiltes, maar spreekt ook langer, waardoor de ratio hetzelfde blijft. Om zeker te weten dat ik met deze data verder kan, heb ik ook de speech van Obama die ik eerder al heb gebruikt geanalyseerd. Hiermee kan ik de theorie niet echt bevestigen, omdat het slechts 3 samples zijn, maar mogelijk wel weerleggen. De gemiddelde ratio was 0.50. Dat is naar mijn idee geen groot verschil. Echter laat Obama in deze speech, in tegen stelling tot in de andere, kortere stiltes, zoals Palin.

Omdat de data zo dicht bij elkaar zit denk ik niet dat de stilteanalyse bruikbaar is om onderscheid te maken tussen een goede en een slechte presentatie. Ik zal de analyse wel uitvoeren om te kijken of de spreker bij de ratio van 0.44 in de buurt komt. Dit lijkt uit deze kleine steekproef de ideale ratio te

zijn. Als er machine learning wordt toegepast, kan dit ideaal worden aangepast op basis van hoe goed een presentatie uit de andere analyses komt.

Nieuwe data: meer stiltes

Bij het refactoren van de code kwam ik er achter dat er in de transcriptie van Sphinx ook twee stiltes na elkaar kunnen zitten. Ik ging uit van steeds één stilte tussen de spraak. Om dit op te lossen heb ik een methode gemaakt die alle aan elkaar grenzende stiltes samenvoegt tot één stilte.

Ook kwam ik er tijdens het refactoren achter dat Sphinx een stilte op twee manieren kan weergeven in de transcriptie: <sil> of </s>. Hierdoor ont er nu meer audio als stilte worden geïdentificeerd en veranderden de verhoudingen wellicht. Daarom voerde ik de analyse opnieuw uit voor Obama en Palin. De resultaten waren als volgt:

	Duur stiltes	SD stiltes	Duur spraak	SD spraak	Gem. ratio	SD ratio
Obama	605.1503759	479.1724154	1485.488722	1226.727179	0.662984971	0.83392249
Palin	440.1557632	641.7502732	1366.697819	1550.391986	0.668074428	1.944692856

De ratio's zijn weliswaar veranderd, maar wederom zijn deze bijna hetzelfde. Ook de meeste andere data ligt dicht bij elkaar. Alleen de standaardafwijking van de ratio is erg verschillend. Een grotere standaardafwijking betekend dat de presentatie minder constant is in de hoeveelheid stilte. Wellicht kan ik hierop toch nog onderscheid maken. Om deze theorie mogelijk te weerleggen analyseer ik wederom ook het tweede bestand van Obama. Het resultaat hiervan is:

	Duur stiltes	SD stiltes	Duur spraak	SD spraak	Gem. ratio	SD ratio
Obama_2	435.0957854	363.703824	1203.524904	1135.61904	0.651767764	0.907814634

Met dit resultaat lijkt het erop dat de standaardafwijking van de ratio inderdaad gebruikt kan worden als voorspellende waarde.

Implementatie

De analyse heb ik voor het voorgaande al geïmplementeerd. Ik heb het in de volgende classes geïmplementeerd:

- `SpeechResultSilence` en `SpeechResultSpeech`. Zoals eerder aangegeven onderdelen van de standaardisatie van het API-resultaat.
- `SilenceAnalysis`. Dit is een implementatie van `ISpeechAspectAnalysis`. Het object hiervan wordt aangemaakt in `SphinxSpeechAdapter` en toegevoegd aan de `ArrayList` met analyses die uitgevoerd moeten worden.

Ik ga uit van dezelfde aanpak als bij de alternatieve analyse. Dit houdt in dat ik voorlopig alleen bepaal waar de ingevoerde audio dichterbij ligt.

Daarnaast behoud ik de functionaliteit die aangeeft hoe dicht de spreker bij de gemiddelden van de drie gemeten waarden bij Obama en Palin zit. Deze functionaliteit zal uiteraard wel de geüpdatete gegevens gebruiken.

Snelheid

De snelheid van spreken is, net als de stiltes, een aspect dat vaak naar voren komt in trainingen. Dat dit een belangrijk aspect is, wordt ook duidelijk uit de volgende bron:

<http://las.sagepub.com/content/18/2/145.short>. Dit is een onderzoek naar de snelheid van praten met de waargenomen persoonlijkheid. Mensen die sneller spreken, worden gezien als bekwamer.

Deze gebruikte grenzen zijn:

- < 110 langzaam
- 110 – 150 gemiddeld
- 150 – 200 snel
- > 200 te snel

Deze grenzen heb ik bepaald aan de hand van de volgende bron:

<http://www.write-out-loud.com/speech-rate.html>

De richtlijn ligt rond de 150, maar omdat in een presentatie ook stiltes moeten worden gelaten, zet ik de richtlijn voor het gemiddelde in eerste instantie op 130. Deze kan nog bijgewerkt worden door de machine learning.

De output van de eerste versie van dit aspect zag er als volgt uit:

Obama

----Speed analysis----

1285 words in 557 seconds.
138.39277700992625 words per minute.

Palin

----Speed analysis----

1114 words in 580 seconds.
115.07868186357219 words per minute.

4.4.5. Tests

Aangezien ik van te voren al vermoedde dat de snelheid en stiltes makkelijker te implementeren zouden zijn, heb ik niet tussendoor getest. Ik had na het implementeren van de stilte-analyse deze uitgebreid kunnen testen, maar het leek mij efficiënter om eerst een dag aan de snelheids-analyse te besteden en dan de beide analyses in één keer te testen.

Testbestanden

De gevonden testbestanden voor het testen zijn:

- Goed: Een TED Talk over de Large Hadron Collider.
- Twijfelgeval: Een college op de universiteit van Boston. Goed verstaanbaar, maar het is een opname van een video-gesprek. Daarom ben ik geïnteresseerd in de uitkomst van de spraakherkenning en analyse.
- Slecht: 'Bad presentation masterclass'. Een video op YouTube die bedoeld is als voorbeeld van een slechte presentatie. Hij is klunzig, onzeker en slecht voorbereid. Dit zijn uiteraard aspecten die niet allemaal in de audio te horen zijn, zeker niet voor de spraakherkenning. Echter verwachtte ik wel een afwijking in stiltes.

Ik heb ervoor gekozen slechts 3 bestanden te gebruiken omdat het de bedoeling is de analyse m.b.v. machine learning te verbeteren. Ik had meer bestanden kunnen gebruiken, maar voor de eerste test leek dit mij voldoende. Ik ben tevreden met deze beslissing. Het had veel meer tijd gekost om bijvoorbeeld 20 bestanden te zoeken, downloaden, om te zetten naar het juiste formaat, te bekijken en in te schatten hoe goed ze zijn, laten beoordelen door de software en de vergelijking te maken.

Testaanpak

Per bestand heb ik de verwachte uitkomst voor elk te analyseren aspect opgeschreven. Vervolgens heb ik het bestand geanalyseerd en de resultaten vergeleken met de verwachtingen.

Als er structureel afwijkingen zaten in de resultaten ben ik nagegaan of het analysealgoritme aangepast moest worden. Daarna ben ik weer van voor af aan begonnen met het testen.

Testresultaten

Hieronder de verwachtingen, uitkomsten en vergelijkingen van deze twee voor elk van de testbestanden.

TED Talk

Verwachting: TED talks staan er om bekend kwalitatief goed te zijn. Ik verwacht dan ook een goed resultaat met misschien een wat hoge spreek snelheid.

Resultaat: Het resultaat viel tegen. Ik had verwacht dat de verstaanbaarheid, gebaseerd op de alternatieven, beter zou zijn. Wellicht komt dit doordat het over een vrij technisch onderwerp gaat en de spreker een redelijk accent heeft. Het nadeel van een technisch onderwerp is dat de spraakherkenning misschien niet alle woorden kent.

Bad presentation masterclass

Verwachting: Bij deze presentatie verwacht ik redelijk goede verstaanbaarheid, maar slechte stiltes en snelheid.

Resultaat: Het resultaat was ongeveer wat ik ervan verwachtte. De verstaanbaarheid was redelijk, maar de stiltes en snelheid waren slecht.

Boston College

Verwachting: Aangezien de spreker een vrij zwaar accent heeft, verwacht ik veel alternatieven en daarmee slechte verstaanbaarheid. Naar mijn idee zijn de stiltes en snelheid prima.

Resultaat: De verstaanbaarheid was redelijk en de score op stiltes en snelheid was goed.

De volledige testresultaten staan in Bijlage E.

Accenten en dialecten

Het accent van de spreker was in twee van de geteste presentaties van vrij grote invloed. Een aspect dat hier op lijkt is een dialect.

De analyse van stiltes en snelheid hebben geen last van het accent of dialect van de spreker. De alternatieven-analyse echter wel. Ik had deze kunnen aanpassen, zodat er (meer) rekening houdt met een accent of dialect. Daarentegen ben ik van mening dat een accent of dialect een presentatie voor een menselijke luisteraar ook lastiger te volgen maakt. Daarom heb ik besloten geen rekening te houden met accenten en dialecten. Ik heb dit punt overeengestemd met mijn begeleider.

Geheugenproblemen

Bij het analyseren van sommige bestanden kwam er een `OutOfMemoryError` voor. Dit was eerder alleen voorgekomen bij het analyseren van muziek. Omdat het alleen bij muziek voorkwam, heb ik het toen gelaten voor wat het was.

Na wat onderzoek blijkt dat het probleem zit in de `StringBuilder` die gebruikt wordt voor het opbouwen van de alternatieven in Sphinx. Intern wordt de `String` bijgehouden als `char[]`. Als de `String` die in de `char[]` moet groter is dan de `char[]`, wordt de lengte ervan verdubbeld. Dit leidt uiteindelijk tot het punt dat het systeem geen geheugen meer kan vinden voor de `char[]`, resulterend in een `OutOfMemoryError`. Er wordt geprobeerd alle alternatieven van een transcriptie in één `StringBuilder` te zetten. Echter, wat ik nodig heb is alleen het aantal alternatieven, niet de alternatieven zelf.

Mogelijke oplossingen

Om dit probleem op te lossen had ik twee mogelijkheden. De makkelijke oplossing is om een try-catch om de code te zetten. Dit zou een snelle, lelijke oplossing zijn. In het geval van de error kan er dan de maximale score worden toegekend. Daarnaast kon ik ook een deel van de code van Sphinx herschrijven zodat deze alleen het aantal teruggeeft.

Omschrijven Sphinx

Ik probeer eerst om de Sphinx code gedeeltelijk om te schrijven. Al snel blijkt dat veel methodes niet functioneren als ik deze kopieer naar mijn eigen package en aanpas. Er wordt gebruik gemaakt van classes en inner classes die ik in hun geheel zou moeten verplaatsen. Om deze classes dan te laten werken, zou ik nog meer moeten verplaatsen naar mijn eigen package.

De reden dat de `OutOfMemoryError` voorkomt is dat alle alternatieven achter elkaar worden geplakt in één enorme `String`. Deze `String` functioneert als queue voor het bepalen van de volgorde van de alternatieven. Het is immers een functie die de *beste* `n` alternatieven terug geeft. Toen ik dit probeerde aan te passen naar een functie die alleen de alternatieven telt, moest ik grote

aanpassingen doen aan de structuur van Sphinx. Dit lijkt mij niet de bedoeling. Ik zal dit nog bespreken met mijn begeleider, maar Sphinx is uiteindelijk slechts één van de API's. Als Sphinx mindere ondersteuning heeft voor de alternatieven, dan zal een andere API dit onderdeel moeten oppakken.

Try-catch

Bij gebruik van een try-catch voor de OutOfMemoryError zal een deel van de data niet gebruikt kunnen worden. Dit resulteert namelijk in een schatting van rond de 14 miljoen voor een fragment van 29 woorden zoals bij Obama. Echter is het ook mogelijk met minder woorden al de error te krijgen. De schatting zou uiterst onbetrouwbaar zijn. Ik denk er over de data die resulteert in een OutOfMemoryError niet mee te nemen in de analyse. Ook dit zal ik overleggen met mijn begeleider.

4.4.6. Bespreking met begeleider

Het grootste punt van verbetering dat hij had, was dat de analyse nu veel te langzaam is. Als ik m.b.v. YouTube de analyses wil verbeteren, moeten deze wel uitgevoerd kunnen worden in enkele seconden of sneller. Een analyse duurt nu ongeveer dubbel de tijd van de audio. Om dit te bereiken gaf hij de tip in de code van Sphinx te duiken en uit te zoeken waarom het zo langzaam is.

Het eerste idee dat ik meteen had, is dat ik de lange fragmenten met veel alternatieven moet overslaan. Deze kosten onevenredig veel tijd en leveren niet veel extra data op. Hiermee denk ik zo'n 30% tijdswinst te behalen.

Een andere mogelijkheid is om niet elk resultaat dat er gegeven wordt te analyseren. De tussentijdse resultaten moeten hoe dan ook uitgeschakeld worden voor als er m.b.v. YouTube een analyse wordt gedaan.

Daarnaast hebben we nog de volgende punten besproken:

De volgende stap is om aan de analyse ook een cijfer te geven. Nu geef ik alleen nog gemiddelden en verschillen e.d. terug. Dit lijkt mij niet al te moeilijk te implementeren, maar het finetunen gaat nog wat tijd in zitten.

Daarnaast moet er een GUI komen. Deze hoeft niet heel ingewikkeld te zijn, maar moet wel YouTube links als audio source kunnen accepteren. De GUI is zo in elkaar gezet, maar een video van YouTube halen en de audio daarvan in het juiste formaat zetten gaat nog wat tijd in zitten.

We hebben afgesproken dat ik dit in één week geïmplementeerd heb. Hij stuurt mij dan 10 YouTube links waarvan ik hem de analyse en cijfers moet terug sturen. Dit is een redelijk enthousiaste deadline, maar ik wil op deze manier een steeds betere inschatting kunnen maken van hoeveel werk iets is.

4.4.7. Versnellen

Het versnellen van de analyse heb ik op twee manieren geprobeerd te doen. Eerst heb ik gezocht naar de bottlenecks in mijn eigen code. Als het dan nog steeds veel te langzaam is, wat waarschijnlijk zo zal zijn, zal ik in de code van Sphinx kijken waar de meeste tijd verloren gaat.

Problemen in analyse

Als eerste heb ik de tussentijdse analyse optioneel gemaakt. Voor het testen is deze m.b.v. een boolean variabele aan en uit te zetten.

Daarnaast sla ik het analyseren van de fragmenten met meer dan 20 woorden over. Dit moet een groot verschil maken. Dit scheelt namelijk miljoenen berekeningen.

Resultaat

Het verwijderen van de tussentijdse analyse gaf niet veel vooruitgang in snelheid. Waarschijnlijk omdat het probleem in het bepalen van het aantal alternatieven zit en dat slechts één keer per fragment wordt gedaan.

Het overslaan van fragmenten langer dan 20 woorden heeft het proces wel redelijk versneld. Dit is zo'n 50%.

Problemen in Sphinx

In de configuratie van Sphinx ben ik gaan zoeken naar instellingen die de performance kunnen beïnvloeden. Ik heb ten eerste de Cache van het Language Model vergroot. Dit gaf in eerste instantie nog maar een kleine vooruitgang. Het probleem hiermee is dat de GarbageCollector out of memory kan gaan bij het opruimen van de cache.

Een andere instelling in de configuratie, de 'beamwidth', bleek enorme invloed te hebben op de snelheid. Dit gaat echter wel ten koste van de kwaliteit van de transcripties. Ik heb geprobeerd deze waarde zo klein mogelijk te maken, dit resulteert in hogere snelheid zonder al te veel aan kwaliteit van de analyse in te leveren. Door het aanpassen van deze instelling bestaan de resultaten uit meer korte fragmenten. Dit heeft als gevolg dat de data die ik tot dan toe gebruikte niet meer vergelijkbaar is. Ik heb met de nieuwe instellingen de speeches van Obama en Palin opnieuw geanalyseerd.

OutOfMemoryError

Een groot voordeel van deze kortere resultaten is dat de OutOfMemoryError niet meer voor zal komen. Het is in theorie nog wel mogelijk, maar ik kan met zekerheid stellen dat het bij normaal gebruik niet zal gebeuren. Het probleem deed zich namelijk alleen voor bij miljoenen alternatieven voor fragmenten van meer dan 20 woorden. Nu zijn er over het algemeen niet meer dan 500 alternatieven en niet meer dan 15 woorden per fragment.

4.4.8. Becijferen

Voor het becijferen maak ik een centrale class die alle cijfers bijhoudt: AnalysisEvaluation. Ook kan deze class een algemeen cijfer genereren op basis van een gewicht dat aan elk cijfer wordt gegeven. Dit is een singleton class om te voorkomen dat er op meerdere plaatsen cijfers worden bijgehouden. Dit heeft wel als nadeel dat je niet meerdere analyses achter elkaar kan doen en de cijfers kan vergelijken of iets dergelijks.

De gewichten die aan de cijfers gekoppeld worden, bepaal ik aan de hand van een aantal testbestanden. Dit zijn de bestanden die ik eerder ook al heb gebruikt.

Eerder in dit verslag, en ook in het plan van aanpak, was één van punten van aandacht bij het maken van de analyses hoe een score geïnterpreteerd moet worden. Een stilte van 260ms met een afwijking van 20ms is lastig te relateren voor de gebruiker. Daarnaast moest er aangegeven worden wat acceptabele waarden zijn en wat de streefwaarden zijn. Met de implementatie van de becijfering, zijn deze punten geen problemen meer. De relativering is vrij duidelijk aangezien het een cijfer tussen 0 en 10 is. De streefwaarde is uiteraard een 10. Ten slotte is de acceptabele waarde afhankelijk van de gebruiker. Een ambitieuze gebruiker gaat misschien voor de 9 of 10, terwijl een ander tevreden is met een 6 of 7.

4.4.9. GUI

De GUI zal de analyses aanroepen en als deze klaar zijn uit de AnalysisEvaluation de resultaten halen en deze weergeven.

Ik heb gekozen voor een simpele GUI met de volgende elementen:

- Een Textbox voor het invoeren van een YouTube-link.
- Een Button voor het selecteren van een bestand.
- Een Button voor het starten van de analyse.

4.4.10. Hercoderen audio

Op het moment kan er alleen nog een bestand geselecteerd worden in het juiste formaat, de juiste bitrate en samplingrate. Om een YouTube-link als input te accepteren, moet ik eerst een audiobestand kunnen hercoderen. Welk formaat er ook in gaat, er moet een .wav-bestand uit komen met 16k bitrate en samplingrate. Ik heb hier een library voor, JAVE, die ik ook voor het hercoderen van de audio voor de Google Speech API heb gebruikt.

Met de tutorial van deze library op de website van Java heb ik vrij gemakkelijk het hercoderen kunnen implementeren. Dit zou ook moeten werken met een video als input, maar dit heb ik nog niet getest.

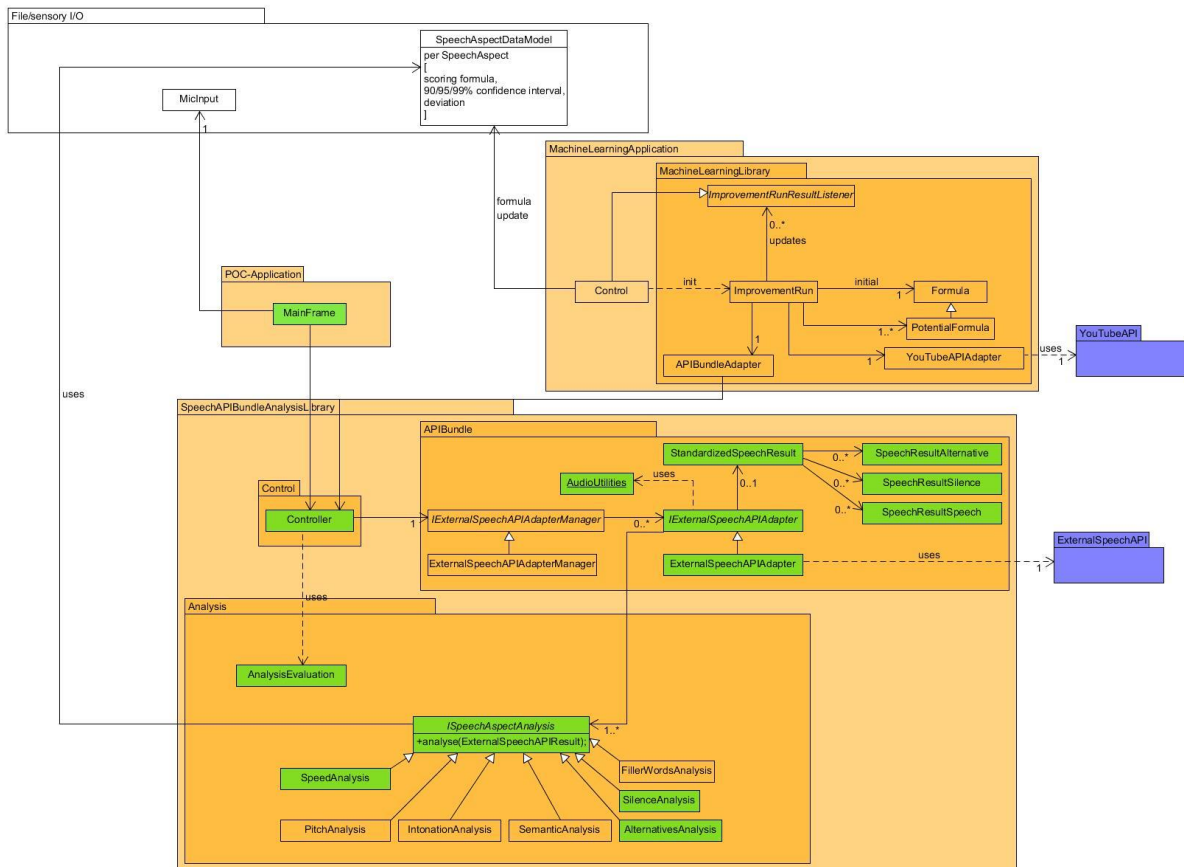
Punten voor verbetering

Op het moment heb ik meerdere methodes in AudioUtilities voor het hercoderen naar verschillende formaten. AudioUtilities zou dit zelf moeten kunnen bepalen op basis van de gebruikte API. De bitrate, samplingrate en bestandsformaat zullen worden opgeslagen in de API-adaptor en meegegeven worden aan de algemenere methode voor het hercoderen.

Dit hercoderen was niet een eis van de opdrachtgever, maar een eigen toevoeging die mij tijd heeft bespaard en ook het gebruik van de library makkelijker maakt.

4.4.11. UML update

Na het implementeren van het becijferen, de GUI en het hercoderen van de audio, ziet het UML diagram er als volgt uit:



De belangrijkste aanpassingen zijn de toevoegingen van de twee nieuwe analyses, AnalysisEvaluation en de GUI. Daarnaast is de code voor Google er uit gehaald.

Wederom is de volgende kleurcodering toegepast. De groengekleurde onderdelen zijn geïmplementeerd. De blauwgekleurde onderdelen zijn externe libraries die ik niet zelf implementeer. Alle oranje onderdelen moesten nog geïmplementeerd worden.

4.4.12. YouTube API

De YouTube API kan statistieken over video's op YouTube opvragen. De API kan echter niet een video downloaden. Voor het downloaden zal een omweg gevonden moeten worden. Het downloaden van YouTube video's is namelijk tegen de servicevoorwaarden van YouTube. Uit de servicevoorwaarden:

- L. Het is u niet toegestaan om Gebruikersvideo's (zoals hieronder gedefinieerd) te benaderen om een andere reden dan uitsluitend uw persoonlijke, niet-zakelijke gebruik, zoals bedoeld met behulp van en toegestaan op grond van de normale functionaliteit van de Dienst, of uitsluitend voor streaming, hetgeen de gelijktijdige digitale verzending betekent van het materiaal door YouTube via internet naar een door een gebruiker middels internet toegepast apparaat, en wel op zodanige wijze dat de gegevens bedoeld zijn om realtime te bekijken en niet om (al dan niet permanent) door de gebruiker te worden gedownload, gekopieerd, opgeslagen of verder te worden verspreid.

Een mogelijke oplossing hiervoor is het streamen van de video naar de analyse library. De enige mogelijkheid die ik zie zonder de voorwaarden te schenden is een stream i.p.v. een download te gebruiken. Ik zal dit bespreken met mijn begeleider.

4.5. Timebox 5 – Testen, refactoren en paralleliseren

4.5.1. Planning

Voor deze timebox stond het volgende op de planning:

- Analyseren 10 bestanden
 - o Verschillen in analyses onderzoeken
 - o Opoffering performance voor consistentie
- Refactoren code
- Begin maken aan de implementatie van de intonatie-analyse

4.5.2. Analyses tien bestanden

De tien links/bestanden die ik van mijn begeleider heb gekregen zijn:

- Carl Sagan's Cosmos: Episode 1 - The Shores of the Cosmic Ocean. Een verhaal over de kosmos vanuit een 1980 perspectief.
- Death to Pennies. Een typische YouTube-video. Duidelijk, to-the-point en snel.
- Douglas Adams: Parrots the Universe and Everything. Een presentatie over zijn reizen. Dit is opgenomen van grote afstand.
- RSA Animate - Drive: The surprising truth about what motivates us. Een typische YouTube-video zoals Death to Pennies. Ik verwacht dat deze ongeveer dezelfde score zullen hebben.
- Sean Plott - My Life of StarCraft. Een gamer over zijn leven rond het spel StarCraft.
- David Blaine: How I held my breath for 17 min
- Hitchens. Een debat/betoog tegen religie.
- Why Is Your BOTTOM in the MIDDLE? Nog een typische YouTube-video. Ook deze zal denk ik ongeveer dezelfde score krijgen.
- Old Spice. Een reclame.
- Pitch Perfect. Muziek.

De resultaten zijn te vinden in bijlage G.

Opmerkingen

De eerste die opvalt, is The Shores of the Cosmic Ocean. Het is op alle aspecten een goede presentatie, maar krijgt een -0.78. Het probleem zit waarschijnlijk in de achtergrondmuziek. Echter is het een documentaire en geen presentatie, het onderscheid van wel of niet een presentatie is uitgebreider beschreven onder 3.3.1. Wat is een presentatie?. Dus wat dat betreft is het goed dat dit het resultaat is.

Daarnaast hebben de video's 'How I held my breath for 17 minutes' en 'Douglas Adams- Parrots the Universe and Everything' een vrij lage score. Van de breathholding-video is hoofdzakelijk de verstaanbaarheid slecht. Ik vind hem zelf af en toe slecht te verstaan, maar een 2.08 is erg laag. Bij Douglas Adams ligt het er waarschijnlijk aan dat het van grote afstand is opgenomen.

Alles bij elkaar heeft het algoritme nu zo'n 9 uit 10 juist beoordeeld. Dit is naar mijn idee goed genoeg om op voort te bouwen.

4.5.3. Verschillen in resultaat van zelfde bestand

Nadat ik de bovenstaande tests had gedaan, heb ik ze in volgorde gezet van goed naar slecht. Om te controleren dat ik geen fout heb gemaakt bij dit copy-pasten, heb ik een aantal bestanden nog een keer geanalyseerd. Er bleek echter dat voor dezelfde bestanden niet altijd hetzelfde cijfer wordt gegeven. Hieronder een voorbeeld.

Cijfers voor de video over het adem vasthouden:

Alternatives score: 11.27206170229426
Silence score: 6.084926800852955
Speed score: 8.398772221511702
Total score: 8.319333066475789
Time taken: 77

Alternatives score: 10.248492678725238
Silence score: 6.878633486264999
Speed score: 8.302852027113442
Total score: 8.320358001043644
Time taken: 73

Alternatives score: 8.303711533944094
Silence score: 6.172011903206629
Speed score: 8.526665814042715
Total score: 7.265731003587689
Time taken: 75

Alternatives score: 9.99260042283298
Silence score: 6.727862136920934
Speed score: 8.526665814042715
Total score: 8.176989453778175
Time taken: 73

Alternatives score: 9.99260042283298
Silence score: 6.856757477054343
Speed score: 8.51067911497634
Total score: 8.239438786461582
Time taken: 74

Dat er überhaupt een verschil is, begrijp ik niet. In principe voer ik precies dezelfde berekening meerdere keren uit. De meeste cijfers liggen redelijk dicht bij elkaar, maar een afwijking van 3, tussen de 11.3 en de 8.3, vind ik erg groot. Opvallend is wel dat er ook twee keer precies hetzelfde uit komt voor de alternatieven.

Grens consistente resultaten

Om er achter te komen hoeveel performance het zou kosten om wel consistente resultaten te krijgen, heb ik gezocht naar de grens. Dit heb ik gedaan door steeds de instellingen van Sphinx aan te passen. Ik doe dit aan de hand van de instelling die mij in eerste instantie performance opleverde. Deze instelling is de `absoluteBeamWidth`. Door deze groter te maken, wordt de analyse preciezer, maar wel langzamer. Bij deze resultaten gaat het er niet om wat de uitkomsten zijn, maar om hoe consistent deze zijn. Als ik de juiste instellingen heb gevonden, zal ik opnieuw de data voor Obama en Palin moeten genereren.

De tests zijn gedaan op basis van de eerder gebruikte “Death to pennies”. Dit is één van de kortere van de testbestanden. Ik heb de huidige situatie als beginsituatie genomen. Vervolgens heb ik steeds de configuratie aangepast en de test opnieuw gedraaid. Uiteindelijk bleek dat Sphinx geen consistente resultaten produceert, ook niet met instellingen die dit zouden moeten garanderen.

Ik besluit hierom het daarbij te laten. Er kan in de toekomst altijd een andere API gebruikt worden die wel consistente resultaten levert.

De aanpak en output van deze tests zijn te vinden in Bijlage C.

4.5.4. Bespreking met opdrachtgever

Ik heb een kort gesprek gehad met de opdrachtgever. Ik heb hem hierin op de hoogte gesteld van de voortgang. Hij had een aantal suggesties-wensen:

- Live spraakherkenning
- Ondersteuning Nederlands
- Export van data naar CSV-bestand voor het genereren van realtime grafieken

Ook stelde hij voor dat ik een presentatie doe bij SMART-Technologies na ongeveer 2 weken om de software te laten zien. Dit is er uiteindelijk niet van gekomen. Ik had graag een presentatie in het Engels met de software aan gegeven, maar ik had meer tijd nodig voor het implementeren van andere functionaliteit en het bijwerken van mijn verslag.

De live spraakherkenning had grote prioriteit. Niet per se omdat ik daar een demo mee kan geven, maar omdat dit het oorspronkelijke doel van de opdracht is.

De ondersteuning van het Nederlands heeft mindere prioriteit. Dit is een lastig punt voor Sphinx. Waarschijnlijk zal een andere library of API die in de toekomst wordt toegevoegd dit voor zijn rekening nemen. Dit heb ik mondeling overeengestemd met de opdrachtgever.

Hiermee kwam de lijst van requirements op het volgende:

Requirement	Prioriteit
De library moet geschreven worden in Java	Must
De library moet Nederlands ondersteunen	Should
Verstaanbaarheid analyseren	Should
Snelheid analyseren	Could
Stiltes analyseren	Could
Aantal keer ‘eh’ tellen	Could
Verloop toonhoogte analyseren	Could

Ondersteuning voor live presentaties	Should
Export van resultaten naar CSV	Should

4.5.5. Bespreking met begeleider

Voor de volgende bespreking met mijn begeleider moest ik mijn code gerefactord hebben. Ik heb het redelijk bijgehouden, maar moest er wel nog een aantal keer goed naar kijken.

Feedback refactoren

Zijn feedback was in grote lijnen:

- Tijdelijke code
- Relatief veel declaraties
- Naamgeving kan hier en daar beter
- Sommige methoden zijn te lang

Ik vond de hoeveelheid verbeterpunten nog wel meevallen. De tijdelijke code is zo verwijderd. Naamgeving is zo veranderd. Alleen het inkorten van methoden was hier en daar een lastige. Er zitten methoden tussen die één grote berekening zijn. Soms is het lastig om deze op te breken in meerdere kleine berekeningen.

In mijn plan van aanpak heb ik staan dat ik wekelijks het hele product zou reviewen. Uiteindelijk zou dit grotendeels tijdverspilling zijn geweest. Door op deze manier een aantal keer met mijn begeleider naar te kijken en zelf te refactoren tussendoor, is er toch nog een soort van reviewen toegepast.

Feedback analyse

Daarnaast heeft hij ook even gekeken naar de beoordelingen van de 10 bestanden en het zoeken van de grens tussen consistente en inconsistente resultaten. Hij had hier de volgende feedback op:

Het is waarschijnlijk niet nodig de daadwerkelijke waarde weer te geven als de score voor een aspect negatief is. Een score van -166 voor stilte zoals bij de Pitch Perfect video zegt niet zo veel. Ook kan ik waarschijnlijk detecteren wat voor een bestand het wel is als het geen presentatie is. Een documentaire geeft heel andere data dan een liedje.

Vervolgstep

Ten slotte hebben we het gehad over de vervolgstap. Op basis van de resultaten die tot dan toe behaald zijn, waren we allebei van mening dat ik het beste de functionaliteit die er toen was kon perfectioneren. Vervolgens kon ik de data-analyse m.b.v. YouTube gaan doen.

Het punt met de YouTube Terms and Conditions, is niet zo'n groot probleem. Het streamen van de video is voldoende. Zolang deze maar geanalyseerd kan worden. Het streamen, en daarmee de analyse, kan dan echter niet sneller dan realtime. Om toch veel data tegelijk te kunnen analyseren, moest ik het mogelijk maken parallel meerdere analyses tegelijk uit te voeren. Ik heb dit aan de requirements toegevoegd

4.5.6. Parallel analyseren

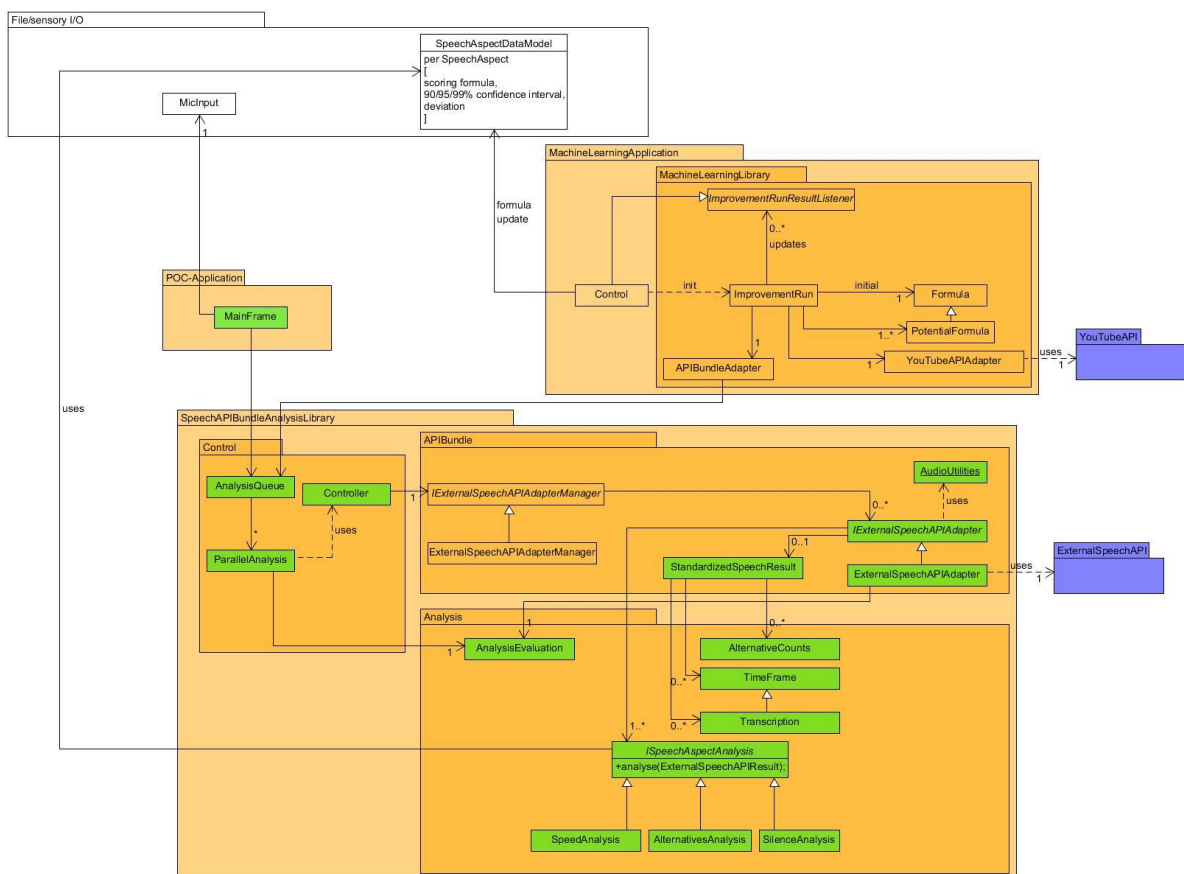
Om de analyses parallel uit te voeren, moest er het één en ander aangepast worden.

Ten eerste moet de AnalysisEvaluation, de beoordeling van een analyse, apart bijgehouden worden voor elke aparte analyse. Deze class was singleton.

Als het parallel analyseren eenmaal werkt, moet er iets van een queue komen. Als ik 1000 bestanden in een uur wil analyseren, kan ik deze niet allemaal tegelijk doen. Ik heb immers geen ongelimiteerd geheugen. Ik zal qua geheugen een maximum van ongeveer 30 parallele analyses aan kunnen, maar bij 5 zit de processor al constant op 100%. Bij 15 parallele analyses wordt dit zelfs minder. Het lijkt er, na wat tests, op dat het geen zin heeft om meer dan 10 parallele analyses te doen. Daarnaast is de tijd die er nodig is om de hele queue te verwerken even lang bij 10 parallele analyses als bij 5. Ik heb besloten er maximaal 5 tegelijk te doen.

4.5.7. UML update

Na de toevoeging van het parallel analyseren en het refactoren is er redelijk wat veranderd aan het classdiagram. Veel benamingen zijn algemener, zoals Transcription i.p.v. SpeechResultSpeech. Ook heb ik threading geïmplementeerd. ParallelAnalysis is een Runnable. Er wordt een wachtrij van bestanden in AnalysisQueue bijgehouden. De Controller class is nog altijd verantwoordelijk voor het aanroepen van de *IEternalSpeechAPIAdapter*, in dit geval Sphinx. Het resultaat van zo'n aanroep is een AnalysisEvaluation object. Hierin worden verschillende cijfers en wegingen bijgehouden.



Wederom is de volgende kleurcodering toegepast. De groengekleurde onderdelen zijn geïmplementeerd. De blauwgekleurde onderdelen zijn externe libraries die ik niet zelf implementeer. Alle oranje onderdelen moesten nog geïmplementeerd worden.

4.6. Timebox 6 – YouTube API

4.6.1. Planning

Voor deze timebox stond het volgende op de planning:

- Implementeren YouTube API
- Refactoren code
- Feedback van begeleider verwerken

4.6.2. Ted Talks download

Ted Talks zijn vrij te downloaden. Ze zijn zelfs individueel via een URL te benaderen. Mijn eerste gedachte was dat er dan wel ergens een download van alle Ted Talks te vinden is. Na wat zoeken vond ik een bestand met daarin 1949 downloads. Dit heeft meerdere dagen in beslag genomen, maar ik kon uiteraard wel verder werken terwijl de downloads bezig waren.

Van Ted Talks zijn vrijwel altijd video's op YouTube te vinden. Door de bestanden ergens anders te downloaden ga ik niet tegen de YouTube Terms and Conditions in, maar kon ik wel de statistieken van YouTube gebruiken voor de machine learning. Op basis van de titel van het bestand kan ik de video vinden op YouTube.

4.6.3. YouTube API

Om gebruik te maken van de YouTube API is er een API-key nodig. Voor nu heb ik deze op mijn eigen Google-account aangemaakt. Ik heb deze in een variabele gezet zodat deze makkelijk vervangen kan worden. De YouTube API die ik gebruik is de YouTube Data API. Hiermee kun je bij data over video's, channels en playlists.

YouTube statistieken

Aangezien de testset op dit punt nog uit alleen Ted Talks bestond, was het naar mijn idee veilig om de volgende aannames te doen:

- De talks staan allemaal op YouTube. Ted en anderen houden dit goed bij.
- Bij het zoeken op de titel van de talk komt de juiste video naar voren. Het zijn namelijk zeer specifieke titels met de naam van de presentator en de titel van de presentatie erin.

Op deze manier kan ik de Videoid van de video ophalen. Aan de hand hiervan kan ik de statistieken van de specifieke video ophalen. Deze statistieken sla ik op in een object en kan ik eventueel in een CSV-bestand opslaan.

Koppeling met analyses

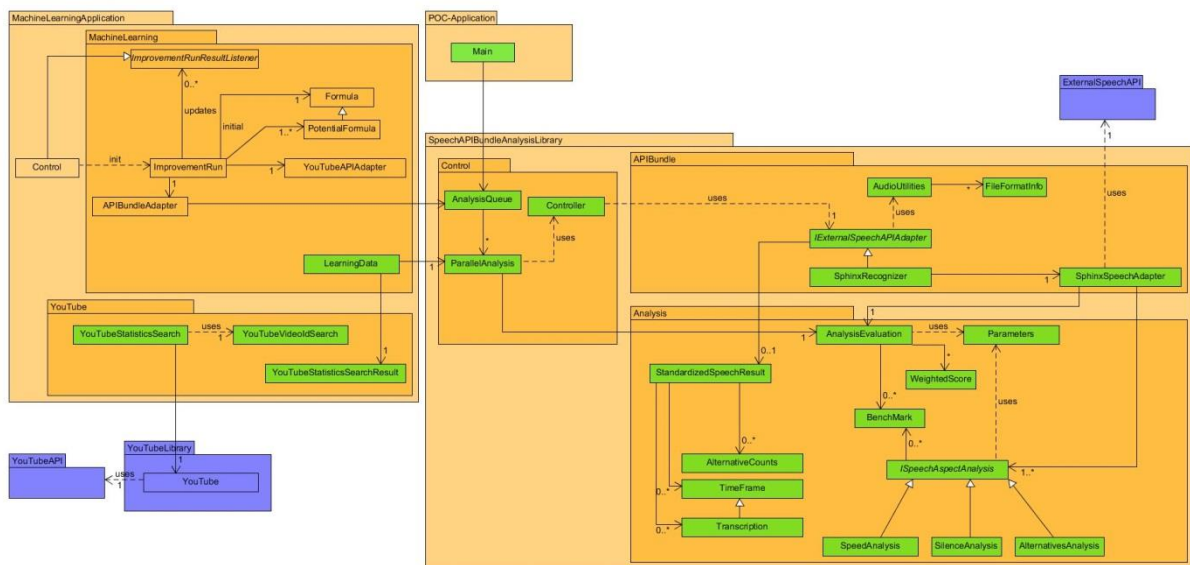
Het object waar de analyse in is opgeslagen wordt samen met de statistieken van YouTube opgeslagen in één object; LearningData. Zo staat alle data die nodig is voor de machine learning in één plaats.

4.6.4. UML update

Bij het refactoren heb in de meeste gevallen van variabelen die bij elkaar horen, een nieuwe class gemaakt voor deze variabelen. Een voorbeeld hiervan is `WeightedScore`, een class met de scores, gewichten en berekeningen die daarbij komen kijken.

Ook is de standaardisatie van het resultaat van een API naar de Analysis-package verplaatst. Dit had eigenlijk al eerder moeten gebeuren, maar heeft mij in geen manier dwars gezeten.

Daarnaast heb ik de YouTube API geïmplementeerd. Dit heb ik in de MachineLearning-package gedaan omdat de analyse de YouTube API niet gebruikt.



Wederom is de volgende kleurcodering toegepast. De groengekleurde onderdelen zijn geïmplementeerd. De blauwgekleurde onderdelen zijn externe libraries die ik niet zelf implementeer. Alle oranje onderdelen moesten nog geïmplementeerd worden.

4.7. Timebox 7 – Afronding analyses, voorbereidingen machine learning

Voor deze timebox stond het volgende op de planning:

- Bijwerken ontwerp document
- Afronden analyses
- Voorbereiden van de code op het implementeren van de Machine Learning
 - o Exporteren van de data over de bestanden in een CSV-bestand of iets dergelijks opslaan. Dit om een bestand geen tweede keer te hoeven analyseren

4.7.1. Analyses afronden

Ik ben alle analyses nog een keer helemaal nagelopen op fouten. Ik heb hier en daar nog het één en ander gerefactord, maar was verder tevreden met het resultaat tot dan toe.

4.7.2. Voorbereidingen voor Machine Learning

Bij het afronden van de analyses zat ik nog met een aantal vragen voordat ik aan machine learning kon beginnen.

- Welke parameters van de analyses moeten bepaald worden door de machine learning?
- Welke parameters verschillen per API?
- Zijn er nog variabelen die mogelijk wel een voorspellende waarde hebben maar die nu niet meegenomen worden in de analyse?

Om de machine learning één centraal punt te geven voor de parameters, heb ik alle parameters die op dat moment werden gebruikt om een analyse op te baseren in één class gezet. Deze class, Parameters, initieert de variabelen en kan op basis van een String één waarde teruggeven.

Er is een aantal variabelen dat API-specifiek is. Deze variabelen zijn de richtlijnen voor het aantal alternatieven dat geretourneerd zal worden. Dit is afhankelijk van de kwaliteit en wellicht ook instellingen van de API.

Er zijn redelijk veel variabelen die ik niet mee heb genomen in de analyses. Dit zijn hoofdzakelijk variabelen waarbij er weinig verschil zat tussen goede en slechte presentaties. Deze kunnen wellicht na het implementeren van de machine learning worden toegevoegd.

Data

Om het proces van de machine learning te versnellen, is het handig om de data waarmee deze werkt al te hebben. Hiermee hoeft de machine learning niet te wachten tot de spraakanalyse klaar is. Hier gaat namelijk relatief veel tijd in zitten. Hier ben ik echter deze timebox niet aan toe gekomen en zet het daarom voor de volgende timebox op de planning.

4.8. Timebox 8 – CSV-export en becijfering YouTube

4.8.1. Planning

Voor deze timebox stonden de volgende punten op de planning:

- Data van en naar CSV
- Categoriseren en/of becijferen YouTube-data

4.8.2. CSV

Export

Om de spraakherkenning niet te hoeven uitvoeren tijdens het toepassen van de machine learning, heb ik geïmplementeerd dat de data waar de analyses hun uitkomst op baseren opgeslagen kan worden. Hierbij gaat het om de data die uit de (externe) spraakherkenningslibrary komt. De parameters die bepaald worden door de machine learning staan hier los van. Uiteindelijk wordt op basis van de parameters de analyse uitgevoerd over de opgeslagen data.

Ik heb ervoor gekozen de verzamelde data op te slaan in een CSV in dezelfde folder en met dezelfde naam als het audiobestand. Vervolgens wordt er, als het audio bestand nog een keer geanalyseerd wordt gecheckt of er een CSV-bestand voor dat audio-bestand bestaat. De analyse zal, als het CSV-bestand bestaat, gedaan worden op basis van de data in het CSV-bestand.

Voor het vullen verzamelen van de data die in de CVS moet komen heb ik nieuwe variabelen aangemaakt in de class AnalysisEvaluation. Deze waarden worden gecheckt en zo nodig ingevuld in de analyses. Als er een waarde niet gevuld is, wordt de hele analyse uitgevoerd.

Het zijn in totaal 30 parameters voor alle analyses en berekeningen samen.

Automatische import

Voor het ophalen van de resultaten uit de CSV heb ik één class toegevoegd: ExistingDataEvaluation. In deze class zitten de volgende methoden:

- hasExistingData. Met deze methode kan voor een audiobestand worden opgevraagd of er bestaande data is voor deze audio.
- evaluate. Deze methode retournt de AnalysisEvaluation die het resultaat is van de analyses uitgevoerd over de bestaande data.
- getData. Dit is een private methode die gebruikt wordt om de data uit de CSV te halen.

Binnen de SphinxRecognizer wordt de evaluate-methode aangeroepen. Ik heb de aanroep binnen het Sphinx-gedeelte van de code gedaan omdat er een Sphinx-specifieke parameter nodig is voor de alternatievenanalyse.

4.8.3. Categoriseren en/of becijferen YouTube-data

Om machine learning te kunnen toepassen met YouTube als bron van data, moet er wel een manier zijn om elke video die geanalyseerd wordt te beoordelen. Hiervoor heb ik een vrij algemeen algoritme gebruikt met de volgende eigenschappen:

- Meer views leidt tot een hogere score
- Meer likes ten opzichte van dislikes leidt tot een hogere score

Hierbij wordt er rekening gehouden met hoe lang de video al op YouTube staat. Een video die een dag op YouTube staat is niet vergelijkbaar met een video van jaren oud.

4.9. Timebox 9 – Live presentaties

4.9.1. Planning

In deze timebox heb ik de library zo aangepast dat er ook live presentaties geanalyseerd kunnen worden.

4.9.2. Live presentaties

Live presentaties analyseren is het uiteindelijke doel van de software en was ook een belangrijke eis van de opdrachtgever. Ik heb dit pas zo laat geïmplementeerd omdat het naar mijn idee geen zin heeft om live audio te kunnen opvangen als daar nog geen analyse voor is.

Voor het analyseren van de live presentaties heb ik één enkele classes toegevoegd: `SphinxLiveSpeechAdapter`. Om consistent te blijven in naamgeving heb ik `SphinxSpeechAdapter` hernoemd naar `SphinxFileSpeechAdapter`. In `SphinxLiveSpeechAdapter` wordt in grote lijnen hetzelfde gedaan als in `SphinxFileSpeechAdapter`. Het verschil zit hem uiteraard in de input.

De Sphinx-library regelt welke microfoon er gebruikt moet worden. Ik hoefde dus niet zelf de hardware aan te spreken.

Om praktische redenen heb ik het mogelijk gemaakt om de analyse van een live presentatie te pauzeren. Als er een presentatie van een uur wordt gehouden, moet niet 10 minuten pauze in het midden de score voor het stilte-aspect onderuit halen.

4.10. Timebox 10 – Machine learning en API-documentatie

4.10.1. Planning

De planning zag er voor deze timebox als volgt uit:

- API-documentatie schrijven
- Machine learning toevoegen aan de library

4.10.2. Machine learning

Aangezien ik geen ervaring heb met machine learning, heeft mijn begeleider uitgelegd hoe het machine learning-algoritme in elkaar moet zitten. Ik heb het implementeren wel helemaal zelf gedaan. Het zit als volgt in elkaar:

- De reeds bestaande LearningData linkt een analyseresultaat aan een resultaat van YouTube.
- De Individual class. Een individu kan random parameters voor de analyses genereren op basis van een ander individu. Met deze parameters kunnen vervolgens de analyses worden uitgevoerd. Het ene individu zal dichterbij de data van YouTube in de buurt komen dan het andere.
- De Generation class. Een generatie bestaat uit meerdere individuen, in dit geval 10.000. Deze 10.000 individuen zullen gegenereerd worden op basis van de 25 besten van de voorgaande generatie. Aan het eind van een generatie worden de beste 25 gekozen om de volgende generatie op te baseren. Zo komt het beste individu van een generatie steeds iets dichterbij de ideale parameters.
- De Monitor class. Deze class beheert de generaties en houdt bij welke individuen van de laatste generatie het best presteerden. Uiteindelijk, na duizenden generaties, kan het beste individu gekozen worden.

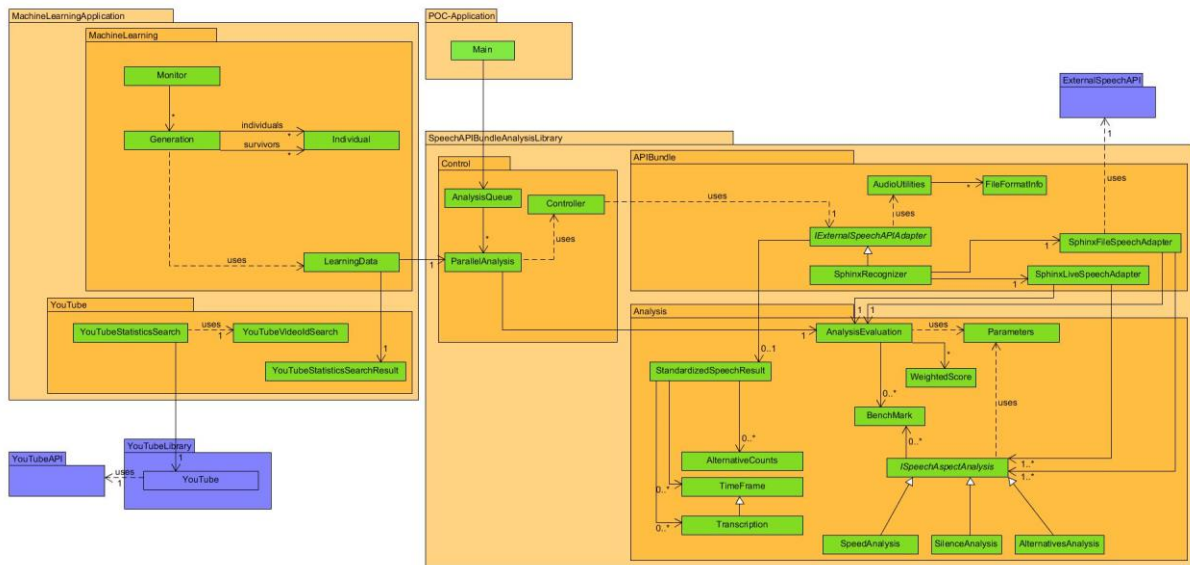
Resultaten

Aangezien ik de machine learning pas in de laatste week heb geïmplementeerd, heb ik geen tijd meer gehad om de volledige 10.000 generaties van 10.000 individuen te voltooien. Ik heb kleinere generaties van 100 individuen wel kunnen voltooien. Na 20 generaties weken de parameters voor de analyses wel af van de oorspronkelijke parameters. De oorspronkelijke en nieuwe parameters zijn te vinden in bijlage L.

Een opvallend resultaat is dat de beoordelingen die als ijkpunt worden gebruikt verder uit elkaar zijn gaan liggen. Daarnaast zijn de wegingen van de verschillende analyses ongeveer hetzelfde gebleven. Dit aan de hand van 2000 analyses van 1949 bestanden. Hieruit kan ik concluderen dat er wel degelijk een verband bestaat tussen de beoordeling die een bestand krijgt van mijn analyses en de beoordeling aan de hand van de YouTube-data.

4.10.3. UML-diagram

Het uiteindelijke UML-diagram ziet er als volgt uit:



Hierin zijn uiteraard ook de aanpassingen voor het analyseren van live presentaties verwerkt.

Alle groengekleurde onderdelen heb ik zelf geïmplementeerd. De blauwgekleurde onderdelen zijn externe libraries of API's.

4.10.4. API-documentatie

In de API-documentatie, die van de library die ik ontwikkeld heb, hoefde ik van de opdrachtgever alleen de classes, public methoden en public velden die nodig zijn voor het gebruik ervan te documenteren. Deze documentatie is te vinden in bijlage I.

Ik heb ervoor gekozen de API-documentatie pas op het laatst te schrijven omdat er nog veranderingen hadden kunnen plaatsvinden door de toepassing van machine learning.

5. Afwijkingen ten opzichte van het afstudeerplan

Er is een aantal noemenswaardige afwijkingen t.o.v. het afstudeerplan.

Het plan was om één spraakherkennings-API te kiezen en die te gebruiken voor de presentatieanalyse. Aangezien er geen API helemaal geschikt was voor het analyseren van een presentatie, is ervoor gekozen de library zo te implementeren dat er meerdere API's tegelijk gebruikt kunnen worden. Voor de proof-of-concept applicatie is wel maar één API gebruikt.

Er is machine learning toegepast om de resultaten van de analyses te verbeteren. Er was in het afstudeerplan geen sprake van machine learning.

De ontwikkelde library is meer richting een proof-of-concept gegaan, dan dat het meteen bruikbaar is bij presentatietraining. Er is namelijk op dit moment nog geen ondersteuning voor Nederlands en het is wenselijk dat er nog meer aspecten worden toegevoegd om een beter onderbouwde beoordeling te kunnen geven.

6. Niet-opgeleverde producten

Ontwerp

Het enige opgeleverde tussenproduct dat niet bij de uiteindelijke producten hoort is het ontwerpdocument. Het plan van aanpak heb ik niet aan mijn begeleider of opdrachtgever opgeleverd. Wel heb ik het mondeling besproken met mijn begeleider. Daarnaast hadden mijn begeleider en opdrachtgever geen behoefte aan testplannen of –rapporten.

In het ontwerpdocument heb ik de veranderingen in het classdiagram bijgehouden. Het complete document is te vinden in bijlage F. De UML-updates zijn ook te vinden onder 4.2.1., 4.4.2, 4.4.11., 4.5.8., 4.6.6. en 4.10.3. Het document geeft inzicht in de aanpassingen die ik heb gedaan aan de structuur van de software en de redeneringen daar achter.

Ik heb dit document vaak achteraf bijgewerkt. Alleen het eerste ontwerp heb ik gemaakt voordat ik de aanpassingen maakte. Ik heb dit zo gedaan om geen tijd te besteden aan het maken van een classdiagram dat later toch weer zou veranderen. Wel wou ik overzicht houden over mijn voortgang. Ook was het makkelijker om een classdiagram te bespreken met mijn begeleider, dan om de code er bij te pakken.

7. Evaluatie

7.1. Aanpak

Zoals beschreven onder 4.2.2. Werkwijze, ben ik vanaf dat moment van werkwijze veranderd. Dit was aan het einde van week 4. De aanpak van herkenbaar falen is mij zeer goed bevallen. Het is een stuk nuttiger om vaker feedback te krijgen op een kleiner deel dan dat je zo lang ergens aan werkt dat je je eigen fouten niet meer ziet.

De timeboxes hadden korter gekund. Ik had wekelijks een afspraak met mijn begeleider. Na dit gesprek moest ik vaak mijn planning voor de tweede week van de timebox aanpassen. 20 timeboxes van 1 week was praktischer geweest. De nieuwe planning was echter snel gemaakt. Ik heb daarmee niet veel hinder ondervonden van de gebruikte aanpak.

Qua werkomgeving had ik waarschijnlijk toch beter op de SMART-afdeling kunnen zitten. Ik had dan waarschijnlijk vaker een gesprek gehad met de heer van Tilborg en meer requirements kunnen verzamelen.

Verder ben ik tevreden met mijn aanpak. Ik heb de planning steeds vrij globaal gehouden. Daarmee kon ik meer tijd besteden aan het echte project en meer werk verzetten.

7.2. Tussenproducten

Onderzoek

Het onderzoek ben ik over het algemeen tevreden mee. Ik ben systematisch te werk gegaan met de longlist-shortlist methode m.b.v. tabellen met voor en nadelen. Vervolgens heb ik de conclusie duidelijk beredeneerd en het geheel duidelijk gedocumenteerd.

Ontwerp

Het eerste ontwerp was veel te uitgebreid, maar daarna heb ik met elke iteratie duidelijke stappen gemaakt. Wat beter had gekund is dat er geen planning van de volgorde van implementatie is te zien in diagrammen. Dit is bij een klassendiagram niet gebruikelijk, maar ik gebruikte nu toch al kleuren om aan te geven wat wel en niet af was.

API-documentatie

De API-documentatie had schematischer gekund. Het is nu een opsomming van classes, methoden en velden met uitleg erbij. Echter is dit naar mijn idee geen probleem aangezien het een kort document is. Het overgrote deel van de library is niet van buitenaf toegankelijk.

Proof-of-concept applicatie

Ik ben tevreden met de proof-of-concept applicatie. Het geeft goed alle functionaliteit van de library weer. Ook is er een applicatie om de machine learning toe te passen.

Machine learning

Ik had eerder aan de machine learning moeten beginnen. Dit is zo gelopen omdat ik dacht dat het lastiger zou zijn dan het uiteindelijk was. Ik zou samen met mijn begeleider er een dag voor gaan zitten, maar uiteindelijk heb ik het zelf in een paar uur in elkaar gezet.

8. Te demonstreren competenties en wijze waarop

Hier zal ik per competentie beschrijven hoe ik deze heb vervuld.

8.1. Selecteren methoden, technieken en tools - niveau 3

Vooraf: De te ontwikkelen software zal gebruik maken van een bestaande spraakherkenningslibrary. Deze library zal aan de hand van een kort onderzoek gekozen worden. Ook zal bepaald worden of het Spring framework en/of nuttig is voor dit project.

Uitvoering

De te gebruiken spraakherkenningslibrary is d.m.v. de longlist-shortlist-methode gekozen. Dit is beschreven onder 4.1.3. en in bijlage B van het afstudeerverslag.

8.2. Ontwerpen systeemdeel - niveau 4

Vooraf: De te ontwikkelen library zal ontworpen worden als herbruikbaar onderdeel van een systeem. De grote lijnen van de structuur van de library zullen bepaald worden, zodat deze herbruikbaar is. De library zal ontworpen worden m.b.v. UML.

8.3. *Uitvoering*

Ik heb m.b.v. UML het systeem ontworpen. De diagrammen zijn te vinden onder 4.2.1., 4.4.2, 4.4.11., 4.5.8., 4.6.6. en 4.10.3., maar ook in bijlage F.

8.4. Bouwen applicatie - niveau 4

Vooraf: Ik zal de library bouwen en om te kunnen laten zien dat de library werkt, zal een proof-of-concept applicatie gebouwd worden. Deze applicatie zal gebruik maken van de library, maar er verder los van staan. Zowel de library als de applicatie zullen gebouwd worden in Java.

Uitvoering

Ik heb de library gebouwd, grotendeels op basis van het UML classdiagram. Het overgrote deel heb ik zelf in elkaar gezet, zonder gebruik te maken van bestaande code. De verschillende classes en de redeneringen achter de verschillende beslissingen zijn te vinden in het ontwerpdocument, bijlage F.

Daarnaast heb ik machine learning toegepast om de resultaten van de analyses te verbeteren. Beschreven onder 4.10.2.

8.5. Initiëren en plannen van het testproces - niveau 3

Vooraf: Aan het eind van elke sprint zal de ontwikkelde software getest worden. Daarbij zullen verschillende testsoorten toegepast worden afhankelijk van wat getest moet worden. Dit zal gedocumenteerd worden in een testplan.

Uitvoering

Testen heb ik uiteindelijk minder aan gedaan dan ik van plan was. Ik heb constant tijdens het ontwikkelen m.b.v. testbestanden de library getest. Daarnaast heb ik twee keer een wat grotere test gedaan. Namelijk:

- In sprint 4 heb ik m.b.v. drie nieuwe bestanden de tot dan toe ontwikkelde algoritmes getest. Ik heb voor elk bestand gedocumenteerd wat de verwachte uitkomst was, wat de werkelijke uitkomst was en daar commentaar op geleverd. Dit is te lezen onder 4.4.5.

- In sprint 5 heb ik 10 bestanden, die mijn begeleider had uitgekozen, geanalyseerd. Hier heb ik echter geen testplan voor geschreven. Ik achtte dit niet nodig. Ik blijf bij deze beslissing.

8.6. Uitvoeren en rapporteren van het testproces - niveau 3

Als afsluiting van een sprint zullen de tests uitgevoerd, gedocumenteerd en gerapporteerd worden.

Uitvoering

De afsluitende tests bij elke sprint heb ik niet uitgevoerd, de reden hiervoor is beschreven onder 4.3.9.

Wel heb ik, zoals bij de competentie “Initiëren en plannen van het testproces” beschreven, tests uitgevoerd. De resultaten zijn op de volgende plaatsen te vinden:

De resultaten van de tests in sprint 4, zoals beschreven onder 4.4.5., zijn te vinden in bijlage E.

De resultaten van de tests in sprint 5 zijn te vinden onder 4.5.2.

9. Geraadpleegde literatuur en andere bronnen

How-to Google Speech API key	http://www.chromium.org/developers/how-tos/api-keys
Bronnen snelheid spreken	http://www.write-out-loud.com/speech-rate.html http://las.sagepub.com/content/18/2/145.short
testbestanden timebox 4	goed: http://www.engr.psu.edu/speaking/SPEECH.html . Ex #1 twijfelgeval: https://www.youtube.com/watch?v=ot6pzc9ZwfU Slecht: : https://www.youtube.com/watch?v=3kgtpI4Q5OY
Jave tutorial	http://www.sauronsoftware.it/projects/jave/manual.php
Sphinx forum	http://sourceforge.net/p/cmuspinyin/discussion/
YouTube API reference	https://developers.google.com/youtube/v3/getting-started
StackOverflow	http://stackoverflow.com/

10. Afkortingenlijst en noodzakelijke, niet algemeen toegankelijke achtergrondinformatie

- Sphinx Sphinx4 voor desktop, niet de beperktere mobiele versie.
- VR Virtual Reality
- IDE Integrated Development Environment
- UML Unified Modeling Language
- Spring framework Framework als uitbreiding op de standaard Java.
- LinkedData Virtuoso Document management systeem
- MoSCoW Het Must-Should-Could-Would principe
- API Application Programming Interface

11. Bijlagen

Bijlage A – Plan van Aanpak

Inleiding

Binnen de afdeling van Ordina die de omscholing en trainingen verzorgt, worden o.a. presentatietrainingen gegeven. Er is bij deze trainingen behoefte aan software die bepaalt hoe goed een presentatie is. Dit zal uiteindelijk gebeuren op basis van de audio en video of andere bewegingsdetectietechnieken. Dit project zal de audio kant van deze analyse verzorgen in de vorm van een Java-library. Er zal ook een proof-of-concept applicatie worden opgeleverd.

De library zal de presentatie beoordelen op basis van verschillende aspecten van verstaanbaarheid. Om dit te bereiken zal er gebruik gemaakt worden van een open-source library voor spraakherkenning. Deze library moet voldoen aan bepaalde criteria en zal gekozen worden aan de hand van een kort onderzoek.

Achtergrondinformatie

Spraakherkenningslibraries hebben over het algemeen mogelijkheden om de snelheid, het aantal verstane en niet-verstane woorden en alternatieven voor verstane woorden te produceren. Ook kan er relatief makkelijk bepaald worden hoeveel stiltes de spreker laat en hoeveel de spreker woorden als 'ehm' en 'uh' gebruikt om tijd te rekken. Deze aspecten kunnen allemaal gebruikt worden om de mate van verstaanbaarheid te bepalen.

Probleemstelling

Zoals reeds beschreven in de inleiding, is er bij de presentatietrainingen van Ordina behoefte aan software die op basis van de audio van een presentatie deze presentatie kan beoordelen. Met zo'n analyse kan objectief feedback gegeven worden over hoe goed een presentatie is. Onder het kopje Probleemanalyse wordt besproken waar hierbij de struikelblokken zitten en hoe deze benaderd zullen worden.

Probleemanalyse

Om een presentatie te kunnen analyseren is er ten eerste library voor spraakherkenning nodig. Aangezien het een open-source library moet zijn, moet er extra aandacht gaan naar het vaststellen van de kwaliteit van de library. Ook de documentatie ervan moet gecontroleerd worden. Een library zonder documentatie levert waarschijnlijk veel tijdverspilling op. Daarentegen kan dit ook geen probleem zijn als de library veel gebruikt wordt en de meeste problemen al een keer zijn opgelost. Het kiezen van de spraakherkenningslibrary zal verder beschreven worden onder het kopje Onderzoek.

Met behulp van deze spraakherkenningslibrary zal een library geschreven worden die de audio beoordeelt. Van deze library moet aangetoond worden dat de waarden die de te ontwikkelen library produceert echt wat zeggen over de presentatie, en niet over de library zelf, en er moet duidelijk worden gemaakt hoe de resultaten geïnterpreteerd moeten worden. Dit aspect wordt verder uitgewerkt onder het kopje Validatie.

Daarnaast moet er antwoord gegeven worden op de vraag hoe de op te leveren library omgaat met accenten en dialecten. Dit zal met de begeleider en/of opdrachtgever besproken worden.

Projectopbouw

Bij het project zullen aspecten van Scrum toegepast worden. Scrum zal niet in zijn geheel worden toegepast omdat er niet in een team wordt gewerkt. Er zal gewerkt worden in sprints van twee weken. Elke sprint zal worden afgesloten met een review van het project. Ook zal een voortgangsrapport opgeleverd worden aan de opdrachtgever en begeleider en een demo worden gegeven aan de stakeholders en andere stagiairs van Ordina. Zoals bij Scrum gebruikelijk is, is er een product owner, Richard van Tilborg. Daarnaast zal er regelmatig overlegd worden met Frank Verbruggen, de begeleider vanuit Ordina.

Er zal gewerkt worden met de Eclipse IDE. Er zal kort onderzocht worden of het Spring framework nuttig is voor dit project. Dit zal ook besproken worden met de opdrachtgever.

Overige personen

De volgende personen zijn betrokken bij het project:

- Frank Verbruggen, begeleider
- Richard van Tilborg, product owner/opdrachtgever
- Martin Weidner, Business Unit Manager
- Ed van Doorn, contactpersoon/begeleider vanuit De Haagse Hogeschool.

Er zullen afspraken met al deze personen gemaakt worden over op welke manier zij op de hoogte willen blijven van de voortgang van het project.

Risico's

Elk project brengt risico's met zich mee. De volgende risico's zijn aan het begin al voorzien en zullen op de daarbij aangegeven manier worden voorkomen of opgelost.

Open-source libraries niet geschikt voor project vanwege beperkte functionaliteit en/of kwaliteit van spraakherkenning.	De bevindingen zullen besproken worden met de begeleider en opdrachtgever. Er zal samen gezocht worden naar een oplossing.
De opdracht neemt meer tijd in beslag dan er is binnen de duur van de stage.	De voortgang zal, ook door de begeleider in de gaten worden gehouden. Als blijkt dat er functionaliteit geschrapt moet worden om binnen de tijd klaar te zijn, zal dit besproken worden met de opdrachtgever.
Er is onvoorzien een probleem met de open-source library waardoor de voortgang misschien grote vertraging gaat oplopen.	Het probleem zal worden besproken met de begeleider. Als de voortgang vertraging oploopt, zie het punt hierboven.

Kwaliteitswaarborging

Om de kwaliteit van het eindproduct te waarborgen, zal elke sprint worden afgesloten met een test van het hele product. Onderdelen die niet testbaar zijn, zoals documentatie, zullen worden gereviewd.

Ook zullen de voortgang en struikelblokken worden besproken met de begeleider en/of de opdrachtgever.

Oplevering

De library zal worden opgeleverd worden met een proof-of-concept applicatie en volledige API-documentatie.

Tijdens het project zal tegen het eind van elke sprint de laatste versie van het product en/of een voortgangsrapport opgeleverd worden aan de opdrachtgever en/of begeleider afhankelijk van hun wensen.

De uiteindelijk op te leveren documenten zijn:

- Plan van aanpak
- Onderzoeksrapport over de librarykeuze
- Lijst met requirements
- Ontwerp van het product
- Voortgangsrapport na elke sprint
- Testplan
- Testrapport
- Spraakanalyselibrary
- API-documentatie
- Proof-of-concept applicatie

Onderzoek

Er zal begonnen worden met een kort onderzoek. Dit zal in eerste instantie niet al te diep in gaan op elke library omdat dat te veel tijd zou kosten, maar uiteindelijk de basis vormen voor de keuze van de open-source library.

Dit onderzoek zal gedocumenteerd worden en opgeleverd worden aan de begeleider en opdrachtgever.

Validatie

De output van de te ontwikkelen library zullen gevalideerd moeten worden. Hoe dit gedaan wordt en de argumentatie hiervoor zullen worden gedocumenteerd en besproken met de begeleider en wellicht anderen.

Er zal een antwoord gegeven worden op de volgende vragen:

- Welke waarden kunnen verwacht worden voor elk gemeten aspect?
- Wat zijn de acceptabele waarden voor elk gemeten aspect?
- Wat zijn de streefwaarden voor elk gemeten aspect?

Deze waarden en eventuele beschrijving en redenering zullen worden gedocumenteerd worden en opgeleverd worden aan de opdrachtgever.

Planning

Er zal gewerkt worden in sprints van twee weken. Aan het eind van elke sprint wordt er een demo gehouden om te laten zien hoe het project vordert.

Start

- Oriënterend literatuuronderzoek naar spraakanalyse
- Kort onderzoek open-source libraries

Iteratief

- Opstellen, prioriteren en overeenstemmen requirements. Dit aan de hand van interviews/gesprekken met medewerkers van Ordina. De requirements worden ingedeeld aan de hand van het MoSCoW-principe.
- Ontwerp van het product. Dit zal gedaan worden m.b.v. UML in zowel analyse- als designdiagrammen
- Bouw van het product
- Refactoren van de code
- Testen. De library zal aan de hand van een testplan per module en vervolgens ook in zijn geheel getest worden.

Afronding

- Overdracht. Hier valt het schrijven van de API-documentatie onder.
- Schrijven van afstudeerdossier

Bijlage B – Onderzoek API's en libraries

Korte samenvatting

Er is onderzoek gedaan naar de beschikbare Java spraakherkenningslibraries. Hiervoor is de Longlist-shortlist methode gebruikt. De gevonden libraries maken allemaal gebruik van drie onderliggende libraries of API's. Deze zijn de Google Speech API, de Java Speech API en de Sphinx4 library. Google is een onbetrouwbare keuze omdat dit niet een officiële API is en Google deze onverwacht kan sluiten. De Java Speech API ondersteunt geen Nederlands. Sphinx4 heeft wel een akoestisch model voor Nederlands, maar geen language model. Dit model zou getraind moeten worden. Er moet tussen deze drie een keuze worden gemaakt.

Inleiding

Voor het ontwikkelen van een spraakanalyselibrary is er behoefte aan een open-source spraakherkenningslibrary. Dit onderzoek zal de basis vormen voor deze keuze. De gekozen library zal als basis voor het ontwikkelen van de spraakanalyselibrary dienen. De keuze zal worden gemaakt m.b.v. de longlist-shortlist-methode.

Dit hele proces wordt vastgelegd in dit document.

Longlist

Voor het opstellen van de longlist zal er gekeken worden naar de volgende aspecten van de libraries:

- Is de library te gebruiken in Java?
- Heeft de library de gewenste functionaliteit?
Hierbij zal worden gekeken naar de aanwezigheid van API-documentatie en of de benodigde functies aanwezig zijn.
- Wordt er actief gewerkt aan de library?
Dit is een indicatie voor toekomstige ondersteuning.

Voor de longlist heb ik zo veel mogelijk verschillende libraries proberen te vinden. Vaak was al snel duidelijk dat een library niet aan de eisen voldeed.

De overgrote meerderheid komt van lijsten van projecten op SourceForge. Op GitHub zijn slechts enkele projecten te vinden. Een deel is gevonden buiten deze twee websites om, maar in alle gevallen stond het project uiteindelijk op een van beide.

SourceForge

Op SourceForge kan gesorteerd worden op os, recentelijk geüpdatet en programmeertaal. Recentelijk geüpdatete Java libraries in de Speech Software-categorie levert de volgende lijst op.

Srt-translator

Dit is een applicatie met als doel het vertalen van ondertitelingen. De onderliggende library is meegeleverd in de download.

De onderliggende service is Google Translate voor de vertalingen en de Google API voor de spraakherkenning. De library lijkt op het eerste gezicht veel talen te ondersteunen.

Er zijn voorbeeld .java-bestanden. Hier zijn geen grote problemen in te zien.

Uit de comments op SourceForge is op te maken dat de library traag kan zijn, dit lijkt echter het enige nadeel te zijn.

Aangezien er software in de download zit, is het makkelijk even een test uit te voeren. De eerste indruk van de STT functionaliteit is redelijk goed. Eén potentieel probleem is dat de software de audio per 2 seconden analyseert en daarmee woorden mist, maar dit is waarschijnlijk een instelling ergens in de code.

Java Speech API

Dit is een implementatie van de Java Speech API-definitie van Oracle. In de documentatie op de website van de library staat: "Needs more work to meet standards". Dit is uiteraard een punt van aandacht, maar de library lijkt alle benodigde functionaliteit te bevatten.

NoiseGator

Software om spraakinput te verwerken en door te sturen naar een virtuele microfoon voor gebruik in andere software. Interessant, maar niet bruikbaar. Het is alleen een uitvoerbaar bestand, de onderliggende library zit er niet apart bij.

Voce

Dit is een library voor voice recognition en synthesis die gebruik maakt van zowel de Java Speech API als Sphinx4. Te gebruiken in Java en C++. Op de website is de complete API-documentatie te vinden. Ook is er een voorbeeld meegeleverd in de download. Het lijkt op het eerste gezicht een goede optie.

HMM Speech Recognition in Java

De library bestaat uit .java-bestanden. Deze zullen in een apart project gezet moeten worden om er een library van te maken. Bij de eerste test, blijkt het resulterende project niet eens te builden. Ook is er geen API-documentatie.

GitHub

Sphinx4 lijkt met duizenden commits tegenover enkele commits de enige actieve library op GitHub te zijn. Sphinx4 is de basis van een aantal andere libraries. Zie het kopje Onderliggende API's en libraries voor de omschrijving.

Ondanks dat de andere libraries niet actief lijken, worden ze wel meegenomen in deze vergelijking.

Jarvis

Deze library maakt gebruik van sphinx4 als onderliggende library. Het project is in de beta-fase en is erg beperkt qua functionaliteit.

Activey/VoiceRecognition

Deze library implementeert alleen de meest basis functionaliteiten van de Google API.

Praat

Deze library heb ik alleen meegenomen omdat deze eerder is besproken met de opdrachtgever. Echter blijkt Praat niet bruikbaar te zijn in Java. Een Java-Wrapper schrijven zou veel werk zijn en daarmee ten koste gaan van te implementeren functionaliteit.

Overzicht

Hieronder een overzicht met in hoeverre de libraries/API's aan de criteria voldoen.

Library/API	Gebruik met Java	Gewenste functionaliteit	Onderhoud library
Srt-translator	V	V	V
Java Speech API	V	?	?
NoiseGator	V	X	
Voce	V	V	V
HMM Speech Recognition	X		
Jarvis	V	X	
Activey/VoiceRecognition	V	X	
Praat	X		

Zoals te zien in bovenstaande tabel, zijn de enige twee libraries die aan alle eisen voldoen de Srt-translator en Voce. Deze worden meegenomen in de shortlist.

Shortlist

Voor het opstellen van de shortlist zal vervolgens uitgebreider worden gekeken naar de libraries. Er zal gekeken worden naar de volgende punten:

- Onderliggende API
- Ondersteuning Nederlands
- Ontwikkelfase van de library

Onderliggende API's en libraries

De onderliggende API's van de gekozen libraries zijn als volgt:

- Srt-translator: Google Translate en Google Speech API
- Voce: Java Speech API en Sphinx4

Ook de libraries die niet aan de longlist-eisen voldeden, zijn gebaseerd op deze libraries. De enige uitzondering hierop is Praat.

Omdat de twee gekozen libraries slechts implementaties van de onderliggende API's zijn, is het goed mogelijk dat de onderliggende API's veel uitgebreider zijn dan de libraries zelf. Ik zal daarom Sphinx4, de Google Speech API en de Java Speech API ook meenemen in de shortlist.

Resulterende lijst

Hieronder een wat uitgebreidere beschrijving van

Voce

Voce maakt gebruik van Sphinx4. Echter ondersteunt Voce lang niet alle functionaliteit van Sphinx4. Bijvoorbeeld, werken met fonetische waarden is niet mogelijk binnen Voce. Ik zie geen reden om een beperktere versie van een uitgebreide library te gebruiken.

Srt-translator – Google Speech API

Niet een officiële API, maar een request naar Google. Dit is naar mijn idee een nadeel aangezien er hiermee geen echte zekerheid is dat deze manier altijd zal blijven werken. Google heeft al een keer, zonder aankondiging, een API dichtgegooid. In dat geval was het de Google Translate API, die nu een betaalde API is. Daarnaast is de Google Speech API niet bedoeld voor dit soort applicaties. Het is bedoeld voor gebruik binnen een plug-in voor Google Chrome. Dit zit nog in de beta-fase.

Srt-translator maakt gebruik van de Google API. Aangezien dit niet een officiële API van Google is, betwijfel ik of het een goed idee is deze te gebruiken. Google heeft immers eerder al de Google Translate API gedeactiveerd. Daarnaast geeft Google niet meer terug dan de tekst die is verstaan en de betrouwbaarheid daarvan.

Sphinx4

Sphinx4 is een goede optie. Op de website staan uitgebreide tutorials om ermee aan de slag te kunnen. Ook is de library qua functionaliteit zeer uitgebreid. Zoals eerder gezegd kan er met de data gewerkt worden tot op het fonetische niveau.

Het grootste voordeel van Sphinx4 is dat het werkt op basis van akoestische modellen. Deze kunnen voor veel talen, waaronder Engels en Nederlands, gedownload worden van SourceForge. Dit zijn reeds getrainde akoestische modellen. Ook kunnen deze modellen nog verder getraind worden.

Een nadeel van Sphinx4 is dat de API documentatie niet elke class, methode en variabele behandelt.

Java Speech API

De Java Speech API library op SourceForge maakt gebruik van de Java Speech API. Dit is voor zover ik heb kunnen vinden de enige implementatie van de API. Dit is niet een implementatie van Oracle of IBM. IBM biedt alleen de documentatie aan. De andere opties die genoemd worden op de Oracle website ondersteunen alleen TTS, Text-To-Speech, terwijl voor het project juist STT, Speech-To-Text, nodig is.

De documentatie van alle functionaliteit is goed. Elke class, methode en variabele is gedocumenteerd.

Het is niet helemaal duidelijk hoe de kwaliteit van deze library is en of deze library Nederlands als taal ondersteunt. Daarnaast staat er in de documentatie al dat er nog werk in gaat zitten om aan de standaard van Oracle te voldoen.

Overzicht

Hieronder een overzicht met in hoeverre de libraries/API's aan de criteria voldoen.

Library/API	Onderliggende API	Functionaliteit	Ondersteuning Nederlands	Ontwikkelfase API
Voce	Java Speech API en Sphinx4	Beperkt	X	Released
Srt-translator	Google Speech API	Beperkt	V	Beta
Sphinx4	-	Uitgebreid	X	Released
Java Speech API	-	?*	?*	Released

Conclusie

De functionaliteit van Voce is matig ten opzichte van Sphinx. Ik zie geen reden om een beperktere API te gebruiken als de onderliggende API ook beschikbaar is.

De Srt-translator is de enige API die Nederlands ondersteund. Echter is de functionaliteit zeer beperkt en is het project nog in de bèta-fase. Daarnaast is er de kans dat Google besluit de API te sluiten.

De Java Speech API is een mindere optie omdat de kwaliteit van de spraakherkenning niet zeker is en het Nederlandse akoestische model nog getraind moet worden. Dit is mij te veel onzekerheid om een project op te baseren.

Sphinx4 is een uitgebreid qua functionaliteit en komt het met getrainde akoestische modellen. Sphinx4 laat dan wel te wensen over in de API documentatie, maar er de tutorials zijn zeer duidelijk. Er zit veel voorbeeldcode bij de tutorials. Ook is het een veel gebruikte library waardoor er veel problemen al opgelost zijn door anderen.

Uit deze en voorgaande punten concludeer ik dat Sphinx4 de beste keuze is.

Geraadpleegde literatuur

Hieronder een tabel met de relevante websites en documenten.

Omschrijving	Link
SourceForge libraries	http://sourceforge.net/directory/audio-video/sound/speech/language:java/os:windows/freshness:recently-updated/
GitHub libraries	https://github.com/search?l=Java&o=desc&q=speech+recognition+library&ref=searchresults&s=stars&type=Repositories&utf8=%E2%9C%93
Srt-translator	http://sourceforge.net/p/srt-tran/code/HEAD/tree/trunk/src/translator/
Java Speech API documentatie	http://www.ling.helsinki.fi/kit/2004s/ctl310gen/L7-Speech/JSAPI/Recognition.html
Sphinx4	http://cmusphinx.sourceforge.net/

Bijlage C – Zoeken naar instellingen voor consistente resultaten

Begin situatie

Dit zijn de instellingen die ik tot nu toe voor de analyses heb gebruikt. Niet één van de analyses geeft consistent dezelfde resultaten.

`absoluteBeamWidth: 200`

Alternatives score: 4.479285454972559
Silence score: 6.195426074885681
Speed score: 8.590612610308224
Total score: 5.851271659346079
Time taken: 85

Alternatives score: 3.9209750784803625
Silence score: 6.797207408790766
Speed score: 8.638572707507354
Total score: 5.948790947263938
Time taken: 78

De resultaten lijken dicht bij elkaar te liggen, maar de individuele resultaten verschillen veel. Dat de uiteindelijke beoordelingen zo dicht bij elkaar liggen is dus toeval.

absoluteBeamWidth verhogen

`absoluteBeamWidth: 400`

Alternatives score: -14.332572172952728
Silence score: 6.3387851693667265
Speed score: 6.9759560046041695
Total score: -1.3333274795983887
Is this a presentation?
Time taken: 108

Alternatives score: -14.332572172952728
Silence score: 6.3387851693667265
Speed score: 6.9759560046041695
Total score: -1.3333274795983887
Is this a presentation?
Time taken: 106

Alternatives score: -14.332572172952728
Silence score: 6.3387851693667265
Speed score: 6.9759560046041695
Total score: -1.3333274795983887
Is this a presentation?
Time taken: 104

Alternatives score: -14.332572172952728
Silence score: 6.3387851693667265
Speed score: 6.9759560046041695
Total score: -1.3333274795983887
Is this a presentation?
Time taken: 103

Deze zijn consistent. Maar voor verificatie draai ik nog een aantal keer draaien dezelfde test. Er is door het verdubbelen van de `absoluteBeamWidth` veel performance verloren gegaan. Wellicht was

dit een te grote stap. Ik zal op basis van of het consistent is/blijft zoeken naar de grens. Bij het zoeken naar deze grens, zal ik steeds twee tests per waarde gebruiken. Pas als ik denk de grens te hebben gevonden, zal ik de test vaker draaien om dit te verifiëren.

absoluteBeamWidth: 300

Alternatives score: -7.08490980478296
Silence score: 6.466873792830938
Speed score: 7.695357462591125
Total score: 1.5385154024457497
Is this a presentation?
Time taken: 95

Alternatives score: -6.387021834167716
Silence score: 6.415196818007753
Speed score: 8.095024939250546
Total score: 1.8243433385973011
Is this a presentation?
Time taken: 91

absoluteBeamWidth: 350

Alternatives score: -9.357762157973575
Silence score: 6.010751034662793
Speed score: 7.66338406445837
Total score: 0.454137716148602
Is this a presentation?
Time taken: 106

Alternatives score: -9.357762157973575
Silence score: 6.010751034662793
Speed score: 7.66338406445837
Total score: 0.454137716148602
Is this a presentation?
Time taken: 102

Alternatives score: -9.357762157973575
Silence score: 6.010751034662793
Speed score: 7.66338406445837
Total score: 0.454137716148602
Is this a presentation?
Time taken: 103

Alternatives score: -9.357762157973575
Silence score: 6.010751034662793
Speed score: 7.66338406445837
Total score: 0.454137716148602
Is this a presentation?
Time taken: 102

absoluteBeamWidth: 325

Alternatives score: -18.968810728853015
Silence score: 6.42509095482159
Speed score: 7.66338406445837
Total score: -2.94283553785179
Is this a presentation?
Time taken: 98

Alternatives score: -18.968810728853015
Silence score: 6.42509095482159
Speed score: 7.66338406445837
Total score: -2.94283553785179
Is this a presentation?
Time taken: 97

absoluteBeamWidth: 310

Alternatives score: -4.363146719383509
Silence score: 6.367522692129772
Speed score: 7.567463870060109
Total score: 2.493514310053584
Is this a presentation?
Time taken: 98

Alternatives score: -3.2465259663991204
Silence score: 6.355783203751226
Speed score: 7.919171249520399
Total score: 2.950340770665993
Is this a presentation?
Time taken: 98

absoluteBeamWidth: 320

Alternatives score: -6.596388225352285
Silence score: 6.514428677911933
Speed score: 7.839237754188515
Total score: 1.763473473722424
Is this a presentation?
Time taken: 99

Alternatives score: -6.596388225352285
Silence score: 6.514428677911933
Speed score: 7.839237754188515
Total score: 1.763473473722424
Is this a presentation?
Time taken: 91

absoluteBeamWidth: 315

Alternatives score: -7.713008978336674
Silence score: 6.40295671766923
Speed score: 7.5834505691264855
Total score: 1.257031313099173
Is this a presentation?
Time taken: 92

Alternatives score: -7.713008978336674
Silence score: 6.40295671766923
Speed score: 7.5834505691264855
Total score: 1.257031313099173
Is this a presentation?
Time taken: 98

Alternatives score: -7.713008978336674
Silence score: 6.40295671766923
Speed score: 7.5834505691264855
Total score: 1.257031313099173

Is this a presentation?
Time taken: 100

Alternatives score: -7.713008978336674
Silence score: 6.40295671766923
Speed score: 7.5834505691264855
Total score: 1.257031313099173
Is this a presentation?
Time taken: 97

Alternatives score: -7.713008978336674
Silence score: 6.40295671766923
Speed score: 7.5834505691264855
Total score: 1.257031313099173
Is this a presentation?
Time taken: 93

Alternatives score: -7.713008978336674
Silence score: 6.40295671766923
Speed score: 7.5834505691264855
Total score: 1.257031313099173
Is this a presentation?
Time taken: 96

Alternatives score: -0.5945516780611939
Silence score: 6.37191761015372
Speed score: 7.711344161657502
Total score: 3.9269199460111
Is this a presentation?
Time taken: 93

Het lijkt in het begin consistent, maar dan zit er opeens een heel ander resultaat tussen. Hoe dit kan is mij een raadsel. Om te testen of Sphinx überhaupt wel consistente resultaten geeft, stel ik de `absoluteBeamWidth` erg hoog in, op 3000. Echter komen er dan nog steeds inconsistente resultaten uit.

`absoluteBeamWidth: 3000`

Alternatives score: -47.52253086419753
Silence score: 7.656257146797651
Speed score: 9.331542731894865
Total score: -2.932757632708794

Alternatives score: -45.7912037037037
Silence score: 7.656356463512945
Speed score: 9.374660893623902
Total score: -2.5749410552341288

Alternatives score: -49.03341049382717
Silence score: 7.656257146797651
Speed score: 9.342322272327124
Total score: -3.2320590145194528

Bijlage D – Transcripts

Hieronder staan de transcripts van de testbestanden. De tekst is grotendeels zonder leestekens, hoofdletters of alinea's om het vergelijken makkelijker te maken. Ook zijn de getallen uitgeschreven. De schuingedrukte tekst was ook voor mij lastig te verstaan. Ik heb <https://www.diffchecker.com/diff> gebruikt om het verschil te bekijken. De nauwkeurigheid heb ik berekend m.b.v. http://www.tools4noobs.com/online_tools/string_similarity/.

Obama

Een opname van de speech van Obama toen Osama bin Laden was gedood.

Handmatig

Aantal woorden: 1393

good evening tonight i can report to the american people and to the world the united states has conducted an operation that killed osama bin laden leader of al qaida and a terrorist who is responsible for the murder of thousands innocent men women and children it was nearly ten years ago a bright september day was darkened by the worst attack on the american people in our history the images of nine eleven are seared into our national memory hijacked planed cutting through a cloudless september sky the twin towers collapsing to the ground black smoke billowing up from the pentagon the wreckage of flight ninety three in shanksville pennsylvania where the actions of heroic citizens saved even more heartbreak and destruction and yet we know that the worst images are those that were unseen to the world the empty seat at the dinner table children who were forced to grow up without their mother or their father parents who would never know the feeling of their child's embrace nearly three thousand citizens taken from us leaving a gaping hole in our hearts on september eleventh two thousand one in our time of grief the american people came together we offered out neighbors a hand and we offered the wounded our blood we reaffirmed our ties to each other and our love of community and country on that day no matter where we came from what god we pray to or what race or ethnicity we were we were united as one american family we were also united in our resolve to protect our nation and the berding to bring those who committed this vicious attack to justice we cli quickly learned that the nine eleven attacks were carried out by al qaida an organization headed by osama bin laden which had openly declared war on the united states and was committed to killing innocence in our country and around the globe and so we went to war against al qaida to protect our citizens our friends and our allies over the last ten years thanks to the tireless *and* heroic work of our military and our counterterrorism professionals we made great strides in that effort we've disrupted terrorist attacks and strengthened our homeland defense in afghanistan we removed the taliban government which had given bin laden and al qaida save haven and support and around the globe we worked with our friends and allies to capture or kill scores of al qaida terrorists including several who were part of the nine eleven plot yet osama bin laden avoided capture and escaped across the afghan border into pakistan meanwhile al qaida continued to operate from along that border and operate through its affiliates across the world and so shortly after taking office i directed leon penetta the director of the cia to make the killing or capture of bin laden the top priority of our war against al qaida even as we continued our broader efforts to disrupt dismantle and defeat his network then last august after years of painstaking work of our intelligence community i was briefed on a possible lead to bin laden it was far from certain and it took many months to run this thread to ground i met repeatedly with my national security team as we developed more information about the possibility that we had located bin laden hiding within a

compound deep inside pakistan and finally last week i determined that we had enough intelligence to take action and authorized an operation to get osama bin laden and bring him to justice today at my direction the united states launched a targeted operation against that compound in abbottabad pakistan a small team of americans with extraordinary courage and capability no americans were harmed they took care to avoid civilian casualties after a firefight they killed osama bin laden and took custody of his body for over two decades bin laden has been al qaida's leader and symbol and has continued to plot attacks against our country and our friends and allies the death of bin laden marks the most significant achievement to date in our nation's effort to defeat al qaida yet his death does not mark the end of our effort there is no doubt that al qaida will continue to pursue attacks against us we must and we will remain vigilant at home and abroad as we do we must also reaffirm that the united states is not and never will be at war with islam i've made clear just as president bush did shortly after nine eleven that our war is not against islam as bin laden is not a muslim leader he was a mass murderer of muslims indeed al qaida slaughtered scores of muslims in many countries including our own so his *demise* should be welcomed by all who believe in peace and human dignity over the years i've repeatedly made clear that we would take action within pakistan if we knew where bin laden was that is what we've done but it is important to know that our counterterrorist cooperation with pakistan helped lead us to bin laden and the compound where he was hiding indeed bin laden had declared war against pakistan as well and ordered attacks against the pakistani people tonight i called president zardari and my team has also spoken with their pakistani counterparts they agree that this is a good and historic day for both of our nations and going forward it is essential that pakistan continue to join us in the fight against al qaida and its affiliates the american people did not choose this fight it came to our shores and started with the senseless slaughter of our citizens after nearly ten years of service struggle and sacrifice we know well the costs of war these efforts weigh on me every time i as commander and chief have to sign a letter to a family that has lost a loved one or look into the eyes of a service member who has been gravely wounded so americans understand the costs of war yet as a country we will never tolerate our security being threatened nor stand idly by when our people are being killed we will be relentless in the defense of our citizens and our friends and allies we will be true to the values that make us who we are and on nights like this one we can say to those families who have lost loved ones to al qaida's terror justice has been done tonight we give thanks to the countless intelligence and counterterrorism professionals who worked tirelessly to achieve this outcome the american people do not see their work nor know their names but tonight they feel the satisfaction of their work and the result of the pursuit of justice we give thanks for the man who carried out this operation for they exemplify the professionalism patriotism and unparalleled courage of those who serve our country and they are part of a generation that has borne the heaviest share of the burden since that september day finally let me say to the families who lost loved ones on nine eleven that we have never forgotten your loss nor wavered in our commitment to see that we do whatever it takes to prevent another attack on our shores and tonight let us think back to the sense of unity that prevailed on nine eleven i know that it has at times frayed yet today's achievement is a testament to the greatness of our country and the determination of the american people the cause of securing our country is not complete but tonight we are once again reminded that america can do whatever we set our mind to that is the story of our history whether it's the pursuit of prosperity for our people or the struggle for equality for all our citizens our commitment to stand up for our values abroad and our sacrifices to make the world a safer place let us remember that we cannot do these things just because of wealth or power but because of who we are one nation under god in the visible with

liberty and justice for all thank you and may god bless you and may god bless the united states of america

Sphinx

Diff: 168 removals.146 additions.

Nauwkeurigheid: 81.89%

good enough tonight i can report to the american people and to the world the united states has conducted an operation that killed or some of them want we're all tired terrorist is responsible for the murder of thousands of medicine man women and children it was nearly ten years ago the right september there was darkened by the worst attack on the american people in our history the images of my own or steered him toward national member i get planes coming through clouds september sky twenty hours collapse into the ground black smoke going up and pentagon the wreckage of floyd ninety three and shag stole pennsylvania for the actions of brooks is saved you more heartbreak than destruction yet we know that the worst images of those who were on sings whirl the empty seat of the dinner table children were forced to rob other mother the father parents would never know the feeling of their child's embrace nearly three thousand services taken from us leaving a gaping hole in our hearts and so tomorrow and two thousand one hard time of grief american people came together we are for neighbors and we offered wounded or blood we we affirmed or tries to each other i'm rowland community and comfort on that day no matter where we came from would god we pray to what race really must be war we were united as one american fan we're also united in our resolve doctor nation and according to bring those who committed this pushes attack to justice the court which we learned that my own attacks were carried out by all kinds realization headed by some of them on which had openly declared war on the united states and was committed to killing innocents in our country and around the globe so we went to war against our pride director citizens our friends and our allies the last ten years thanks to the carlos and roared worker military and our counters fashions we've made great strides metaphor the disrupted terrorist attacks in strength and we're all my indifference afghanistan we removed solomon government which had given them on an all time as a cave in and support the rahman globe we worked with our friends and allies to capture were killed scores of aka terrorists including several who were part and i'm on top it was all the maud newborn get to escape across the afghan war in august they walked by that continue to operate from all that border operator was going to across the world and so shortly after taking office i brightly on intimate the director of the cia to make the killing will capture of them on the top priority on him again stalker even as we continue number our efforts to disrupt dismantled kiki is now then last august after years of pain staking want our intelligence community i was briefed on a possible lead to the mob as far from certain the tuchman monster runners bred to grow i met with beautifully women nationals purity missouri the baltimore information about possibly we located him on hiding with the compound deacon sidewalk years and finally last week i determined that we have enough intelligence to take action unauthorized an operation to get call some of them on and bring him to justice today my direction the united states launched retorted operation against that compound you know bob bob barker stock small team american spirit of the operation with extraordinary courage people don't no mountains walk on they took care to aboard civilian casualties after firefighter the job of some of them on custody of his body for over two decades the mom has been all kinds leader and some this continued applaud attacks against our country and our friends and allies that didn't want marched more significant treatment to date of our nation's effort to defeat pocket is that does

not mockery and over there's no doubt that o'connor will come to pursue attacks against us we must and we will or maybe george at home abroad as we do must also reform that the united states is not been never will be at war with islam i made clear just as president bush detroit astronauts but i was not against islam is the mob was not a muslim leader using mass murderer of muslims and you know kind of slaughtered spores blossoms mary countries including our own so the minors should be welcomed by boldly victories and human dignity or the years i've repeatedly made clear but we would take action with an pockets and if we knew were babylon was that is what we've done but it's important to note that our cover terrorism cooperation with august and help lead us to belong and the compound word was like indeed the mom declared war against pockets on as well in order attacks against the parker's stargate tonight i call prisons are guarding my team is also spoken with their bodies i counterparts they agree that this is a good an stuart a verbal tarnished igor for what it is a central park is targeting you to join us invited him stop fighting phillips the american people did not choose the spot it came to our shores it started with senseless slaughter for service after nearly ten years of service struggle and sacrifice we know well because of war these efforts where are your timeline those men and she felt asylum are true family that has lost a lot look i'm a eyes was service member good rewarded so americans understand the cost of war that is a country we will never tolerate our surgery being right or stan i would buy one of people have been killed we will be relentless in defense of our service and our friends mouths we will be true to the values that make us we are on nights like this we can say to those families were lost loved ones tacos terror justice has been done tonight we give thanks to the countless intelligence encounters with russians were carlos moya she was a outcome the american people do not see their work normal earnings it's like they feel the satisfaction of the war i'm a result of the pursuit of justice give thanks for the manager of this operation were they example five professionals are patriotism and on payroll of courage posters are returned milk for the generation those born heavier sure the burdened sense that september day finally we say that the families who lost loved ones on my watch but we've never forgot your loss nor wavered in our commitment to see that we do whatever it takes to prevent another attack on our shores and sunlight when this thing back to the sense of unity the prevailed on michael i know that there's a time strayed that today is achievement as a testament the greatness of our country and the determination of the american peep cause of securing our country is not complete but tonight we are once again our mind that america can do whatever we saw your mind has the story of our history whether it's the pursuit of prosperity for people for the struggle for equality for all our citizens our commitment to stand up for values abroad a sacrifices to make the world a safer place what is remember that we can do these things not just because of wealth or power because of who we are one nation under god in the visible liberty and justice for all thank you i got bless you may god bless the united states where

Google

Dit is het resultaat van een deel van het bestand. Er zijn nog wat problemen met lege resultaten. De originele tekst van dit deel is 203 woorden.

Diff: 21 removals. 16 additions.

Nauwkeurigheid: 84.79% Dit komt redelijk in de buurt van de confidence die Google terug geeft. Deze is 0.89416337.

good evening tonight i can report to the american people into the world of united states is conducted an operation that killed osama bin laden the leader of al qaeda a terrorist was responsible thousand women and children the right september day was dark and by the worst attack on american people in history images of nine are seared into our national memory hijack planes going through a cloudless sky the twin towers collapse into the ground black smoke billowing up from gallagher flight ninety three in shanksville pennsylvania with actions of peroxidase and saved even more heartbreak the worst images of those the first of the world hearts on september eleventh two thousand one in our time of grief american people came together we offer our neighbours a hand

Lege resultaten

Google geeft soms een leeg resultaat terug. Er lijkt geen patroon in te zitten wanneer dit wel of niet gebeurt. Het is mogelijk dat je een request doet, antwoord krijg, het zelfde request nog een keer doet en een leeg antwoord krijgt. Ik heb geprobeerd de parameters te veranderen, het bestand aan te passen naar een lagere grootte of bestandstype en een vertraging van een minuut op elke request te zetten. Niks lijkt effect te hebben.

Met deze lege resultaten is de Google Speech API niet betrouwbaar. Ik zal er weinig rekening houden. Misschien als de API officieel wordt, kan deze weer gebruikt worden.

Opmerkingen

De nauwkeurigheid van Google is beter en de fouten die gemaakt worden zijn ook kleiner. Bij Google zijn het ook spellingen, zoals neighbour of neighbor. Bij Sphinx zijn het grotere fouten, zoals enough i.p.v. evening. Wat wel handig is aan Sphinx, is dat Sphinx stiltes detecteert en zo bepaalt per hoeveel seconden het bestand geanalyseerd moet worden. Bij Google ben ik genooddakt het zelf in delen van 15 seconden op te delen. Wellicht kan de stiltedetectie van Sphinx gebruikt worden om het bestand op te delen voor Google, maar dit bewaar ik voor als er tijd over is.

Sarah Palin

Dit is een opname van haar speech op de Conservative Political Action Conference 2015.

Handmatig

Woorden: 1258

thank you guys thank you so much i am so honored to get to be here thank you very very much ha thank you im glad to be here and i am so honored tonight to be able to speak about america's finest our veterans i was asked to speak about veterans and i said absolutely talk about near and dear to my heart this subject sergeant dakota meyer thank you so much for that introduction ehm his selfless service reminds us of that that love patriotic love that our country was built upon and his selfless service inspires us he and his generation of vets remind us of that courage that founded this country in fact i'd like to tell you a story about america's first veterans and what this city's namesake said about them seventeen eighty three it was just days after the last british troop left our soil and the father of our country general george washington he gathered his officers in a new york city tavern to bid them farewell these hardened men they fought for years against the greatest military might in the world at that time against all odds they won freedom freedom for themselves and freedom for every american since and standing before them washington was overcome with emotion he was noting the sacrifices that they had made he said with a heart full of love and gratitude i now take leave of you i most devoutly wish that your latter days may be as prosperous and happy as your

former ones have been glorious and honorable that was the wish for our first veterans and it should be ours for every veteran that's followed but we must ask have we lived up to that from then until now there's been a long of heroic vets and they connect our history from george washington to chris kyle millions like them millions like them ordinary americans so equipped for extraordinary things because they have come forward to serve and to save americans lives how many have done more than this generation in uniform though the war on terror is the longest military engagement in u s history so many have served in irak and afghanistan tens of thousands wounded more than sixty eight hundred dead the strain on military and their families it is enormous during world war two the average deployment in the combat theater it was six months korean war nine months vietnam thirteen months for irak and afghanistan an initial enlistment was 45 months these deployment costs they are huge and they're hidden the longer someone's deployed and then redeployed well the more likely they'll suffer p t s d in about half a million of our returning vets they suffer some form of it they suffer disproportionate unemployment numbers and the average divorce rate it's around eighty percent and worse oh friends worse the suicide rate the suicide rate among our best and our brightest is twenty three a day as we gather here we're safe we're secure we're having fun four days together in a conference in these four days ninety two of veterans will have taken their lives so have we lived up to washingtons wish for our vets i don't ask that said the politician as dakota was suggesting there no i i ask as one of so many mothers of a combat vet when my son went off to war the first time as a teenager i was confronted with the same reality that all the other moms have to face and that's realizing that man i wasn't gonna be there to help to protect moms can't be there when they hurt i could pray and i did and i do but oh few things are more difficult than to kiss a child goodbye off into harms way knowing you're not gonna be able there to protect those first deployments ehm that is when a parent goes calling him son to calling m sir america hands over her sons and her daughters and *servants* with the promise that they're gonna be taken care of our troops are promised that no one will be left behind on the battlefield they're promised that a grateful nation will spare no expense to patch them up and bring them back to health when wounded and now they come home wounded too many broken in body and spirit well we their mothers and their fathers and their husbands and their wives we're here to collect on the promises made we can't wait for d c to fix their bureaucratic blunders this bureaucracy is killing our vets they wait for months they wait for years to get treatment at the v a and they're losing hope the v a's mistakes and those cover-ups they have cost the lives of five hundred vets in just the last four years and that doesn't account for those that took their own lives in despair we witness the way a corrupt government treats our vets the v a putting them on those secret waiting lists and deceiving congress and spying on investigators who're just trying to get to the bottom of it and don't be fooled into thinking that the problems resolve simply because the media doesn't cover lame duck scandals anymore just because one guy at the top resigned well the problems didn't resign the reason that you don't hear about them is because the vets don't whine they're not wired to complain that's why this ran under the radar for so long well our debt of gratitude starts payment with three simple solutions that government can and should do right now because it is time to demand solutions first with healthcare give vouchers for treatment outside the v a give our vets the same freedom that they gave us and instead of illegal aliens cutting in line being rewarded with a hand out of u s benefits no we demand that the vets are first in line and second allow vets who are reentering the civilian workforce let them use the skills that they learned in the military today say a vet with superb computer or ehm mechanical skills often they have to go backwards they have to go back and take classes to get a paper degree or paper certificate to slap on the wall that says ok now i'm certified in a field that they

already knew let them test out and their military certification can transfer over common sense yea common sense which i know is an endangered species in these parts yea third secure their benefits congress secure their benefits did you know that last year congress actually voted to cut vets retirement benefits by twenty percent did they vote to cut their own no and they only reverse course when enough of us we rose up in protest so take the issue away from the politicians after legislation to secure benefits permanently now healthcare and benefits you know this is just part of the equation general washington wished for our veterans that their deeds in war would be recognized as glorious and honorable and the deeds have been certainly there's no question but see the thing that those vets from the revolutionary war what they had that today's troops don't is victory and that is the thing that they cherish most they deserve to know that their sacrifices are not in vain to know what they fought for and their friends died for was worth it

Sphinx

Diff: 83 removals.80 additions.

Nauwkeurigheid: 64.62%

and and why i didn't wanna get near the very very much thank you i landed here and i don't want nineteen the evils use the battered by our bedroom and the vow that clinton i think absolutely and i'm hearing here in my heart is that i didn't i liar i think you so much that you're next hang it down remind us oh that that was the strike while the ark and you don't mind him now but there is i resigned v. n. u. generation of that remind us of that courage and found it country in fact i think i used to worry about america's first then turned and went and evening say that about them then he read what you have days after the last word is true that our soil and the father of our country general george washington the hammered him out there in your study camera didn't answer well these hardened and in fact three years and if the glory of military might in the world and now i'm they whine you know i'm breakthrough and down and bring them for every american men and the courthouse was it was overcome with emotion you knowing that that i'd been made he's at when a heart full of love and gratitude i don't see you i'm not devout we wish that your naturally maybe it's not worth unhappy as your former wednesday and ignore your friend honorable what do we or however is that jury and the powers for every afternoon followed the only one half can we do that from that until now is it won't mind what then and they kidnapped our history from george washington do you the decreased cairo no i like them millions like him we're carrying her consulted with links to ruin your evening the car or regent arabs and the american why we have them work in this generation in uniform don't want to hear him don't want military needs making us history don't ever been iraqi a entered the end tens of thousands wounded more than sixteen hundred ad the street and military and their families even warm during world war two the howard's deployment in the contact your view it not korean war nine months vietnam thirteen months we're iraqi in afghanistan and unusual in lisbon with borderline mind the white mankind are huge in their hidden the longer someone's deployed in and read the white woman more like we know that repeats yesterday and i happen to me and i returned that a factory or not it averages the worst employment number and the average of four story it cracking percent and worse over and worse and so what i worry so every time on our behalf that outcry twenty three again every gather here we were thinks you're having line or he's generally concrete in the story of nineteen to one are better and you had taken their life don't happen to want to sweets for that i don't have that politician and the corner with a dent in there no i i can't get one and so many mothers a book on baghdad when i cannot

toward the first time in the teenager i was impressed with the reality only undermined activated that's realizing it what can be their help attack monsignor be there when they heard great maybe i can i do but the house do things are more difficult to get the town good-bye i think i'm arms wait no uniting the wall there those stories deployment and when hearing about them collins that's it ghali names there and that handler even higher than what prompted the media here on our troops are promise that no one will be back behind on the battlefield their product and we will spare no expense jim not been linked to help when we did and how they go and you were looking in body and spirit well we their mothers and their honors in their husbands and their wives were here to pull out on a promise them a we returned the next their bureaucratic one years the bureaucracy is here we are back they wait for mine hey we only a hurry he had been two oh maybe they can come right the costs lives i'm hundred bandages the last four years and that counts for those into their own lives and they are we went away and corrupt government three darden the da putting them on those things that we live and deceiving congress only nine years and is trying to walk on it no undertaking of that problem the result and as mummy and then they have arlene that he had only anymore just because my daughter who's moment what problems did resign they're worried and you hear about it it and i don't want to get what you did you see why everyone and everything was well the gratitude the aren't being with reasonable solution the government can and should you right now because it's time to damien pollution or else here about cures for treatment outside the yay how three hundred eighteen or anything that you really mean that you whine we're in and out of the lead no we demand that servers him on a hap lou the admirably in three months ago you were forced on you the old angered military today they have that right for being here or that mechanical hills when you have warned him to happen to happen to get a burglary beaver certificates of all unfair that the horns are invite you really are new let's get out in their military certainty he transfer over and then why you wouldn't use the scene in these parts carter thank you don't have that congress securing their benefit when you know what i hear you recently voted to cut back its retirement benefit one percent did they don't think that arrow norton the rumors horsemen enough about what was on the protests so you are we from the politician after reading writing and secure benefit permanently now out here and then it's the differences are the inflation general washington wish for our veterans if there dean's in war when you recognize as laurie in an honorable and the dean's poet and certainly there's no question i think that do that the revolutionary war what the hell that these dreams don't in the jury and i think that the aids years most they concerts and all that there's appetite and are not in a the knowledge what did i work and everything behind or what were they

[Google](#)

Lege resultaten, zoals eerder aangegeven.

Bijlage E - Tests

TED TALK

TED talks staan er om bekend kwalitatief goed te zijn. Ik verwacht dan ook een goed resultaat met misschien een wat hoge spreek snelheid.

----Low----

Avg: 1719.8493150684933
StdDev: 6565.161611938245

Difference with Obama:
Avg 1167.1954688684932
StdDev 4789.667459938245

Difference with Palin:
Avg 1032.9860671684933
StdDev 5074.414385938246

Closest to: Palin

----Med----

Avg: 121133.44444444444
StdDev: 169180.64706069138

Difference with Obama:
Avg 74472.60233444444
StdDev 77638.75717069137

Difference with Palin:
Avg 8351.652355555561
StdDev 124390.6306393086

Closest to: Palin

----High----

Avg: 444522.75
StdDev: 137100.5182613391

Difference with Obama:
Avg 555477.25
StdDev 137100.5182613391

Difference with Palin:
Avg 555477.25
StdDev 137100.5182613391

Closest to: Obama

----Total----

Avg: 123312.20645161289
StdDev: 197320.59994517386

Difference with Obama:
Avg 106478.14136161288
StdDev 107833.10342517386

Difference with Palin:
Avg 76329.41837161289

StdDev 1152.658245173865

Closest to: Palin

----Silences analysis----

Average silence: 358.73015873015856

Difference with guideline: 156.1931802698415

Average speech: 2118.3492063492063

Difference with guideline: 697.8210973492062

Average ratio: 0.3224688886490951

Difference with guideline: 0.34329924335090484

Ratio standard deviation: 0.49149102163902647

Difference with Obama: 0.3424314683609735

Difference with Palin: 1.4532018343609736

Closest to: Obama

----Speed analysis----

1974 words in 790 seconds.

149.86334649255997 words per minute.

Speech speed category: medium

Het resultaat viel tegen. Ik had verwacht dat de verstaanbaarheid, gebaseerd op de alternatieven, beter zou zijn. Wellicht komt dit doordat het over een vrij technisch onderwerp gaat en de spreker een redelijk accent heeft. Het nadeel van een technisch onderwerp is dat de spraakherkenning misschien niet alle woorden kent.

Bad presentation masterclass

Bij deze verwacht ik redelijk goede verstaanbaarheid, maar slechte stiltes en snelheid.

----Low----

Avg: 539.2089552238807

StdDev: 1241.4186598442518

Difference with Obama:

Avg 13.444890976119268

StdDev 534.0754921557482

Difference with Palin:

Avg 147.65429267611933

StdDev 249.32856615574815

Closest to: Obama

----Med----

Avg: 5472.0

StdDev: 0.0

Difference with Obama:

Avg 41188.84211

StdDev 91541.88989

Difference with Palin:

Avg 124013.0968

StdDev 293571.2777

Closest to: Obama

----High----

Avg: NaN

StdDev: NaN

Difference with Obama:

Avg NaN

StdDev NaN

Difference with Palin:

Avg NaN

StdDev NaN

Closest to: Palin

----Total----

Avg: 611.75

StdDev: 1369.6526237340227

Difference with Obama:

Avg 16222.31509

StdDev 88117.84389626597

Difference with Palin:

Avg 46371.03808

StdDev 194798.28907626597

Closest to: Obama

----Silences analysis----

Average silence: 552.2602739726028

Difference with guideline: 37.33693497260276

Average speech: 1002.4657534246571

Difference with guideline: 418.06235557534296

Average ratio: 1.2389166941995402

Difference with guideline: 0.5731485621995402

Ratio standard deviation: 2.3047783696824515

Difference with Obama: 1.4708558796824516

Difference with Palin: 0.36008551368245145

Closest to: Palin

----Speed analysis----

276 words in 232 seconds.

71.28712871287128 words per minute.

Speech speed category: slow

Dit is ongeveer wel het resultaat dat ik verwachtte.

Boston College

Aangezien de spreker een vrij zwaar accent heeft, verwacht ik veel alternatieven en daarmee slechte verstaanbaarheid. Naar mijn idee zijn de stiltes en snelheid prima.

Alternatives analysis

----Low----

Avg: 1781.2959770114933

StdDev: 6237.676840346468

Difference with Obama:
Avg: 1228.6421308114932
StdDev: 4462.182688346468

Difference with Palin:
Avg: 1094.4327291114932
StdDev: 4746.929614346469

Closest to: Palin

----Med----
Avg: 107196.02479338842
StdDev: 204734.9655633051

Difference with Obama:
Avg: 60535.182683388426
StdDev: 113193.0756733051

Difference with Palin:
Avg: 22289.072006611575
StdDev: 88836.31213669488

Closest to: Palin

----High----
Avg: 827878.7
StdDev: 271754.98081259814

Difference with Obama:
Avg: 172121.30000000005
StdDev: 271754.98081259814

Difference with Palin:
Avg: 172121.30000000005
StdDev: 271754.98081259814

Closest to: Obama

----Total----
Avg: 45656.36116910229
StdDev: 164628.36940236198

Difference with Obama:
Avg: 28822.29607910229
StdDev: 75140.87288236197

Difference with Palin:
Avg: 1326.4269108977096
StdDev: 31539.57229763802

Closest to: Palin

----Silences analysis----
Average silence: 433.3926981300092
Difference with guideline: 81.53064086999086
Average speech: 1152.2885129118433
Difference with guideline: 268.23959608815676
Average ratio: 0.5410369567950523
Difference with guideline: 0.1247311752049477

Ratio standard deviation: 0.9293933157396631
Difference with Obama: 0.09547082573966315
Difference with Palin: 1.015299540260337
Closest to: Obama

----Speed analysis----
3770 words in 1780 seconds.
127.00801239759909 words per minute.
Speech speed category: medium

De verstaanbaarheid was redelijk en de score op stiltes en snelheid was goed.

Bijlage F – Ontwerp

Inleiding

In dit document zal het proces van het ontwerpen van het systeem worden besproken. Het ontwerp zal worden doorgenomen en de afwegingen bij het ontwerpen zullen worden besproken en beargumenteerd.

Het systeem moet op basis van de audio van een presentatie, met behulp van meerdere API's of libraries, de kwaliteit van de presentatie kunnen bepalen aan de hand van een aantal aspecten. Deze beoordeling moet verbeterd worden aan de hand van beoordelingen door mensen op YouTube. Daarnaast moet er een proof-of-concept applicatie worden gebouwd om het geheel te kunnen demonstreren.

Aspecten

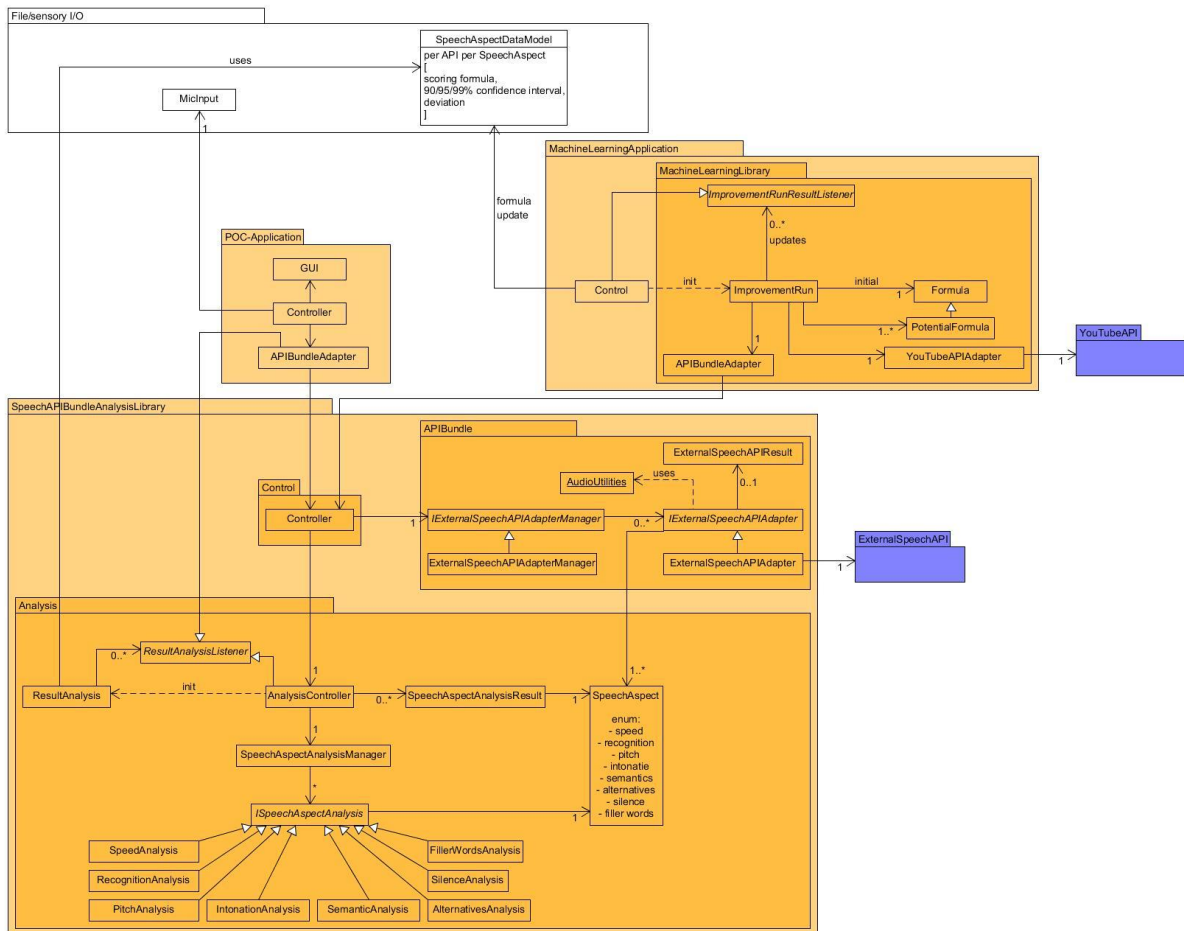
Het is de bedoeling om verschillende aspecten van een presentatie te kunnen meten en beoordelen. Dit zijn de volgende aspecten:

- Snelheid
- Percentage herkende woorden
- Toonhoogte
- Beeldspraak
- Intonatie
- Alternatieven voor verstane woorden
- Stiltes
- Vulwoorden (ehm, like)

Deze aspecten zullen worden gemeten op basis van de output van verschillende API's of libraries. Welke aspecten uiteindelijk geïmplementeerd worden hangt af van hoe vlot dit gaat en welke aspecten belangrijker of geschikter zijn. Met geschiktheid bedoel ik hier het gemak waarmee het aspect m.b.v. een andere API geïmplementeerd zal kunnen worden. Er zijn uiteraard aspecten die niet door elke API berekend kunnen worden.

Eerste ontwerp

Hieronder is het ontwerp afgebeeld in een vereenvoudigd UML klassendiagram.



Argumentatie

De software zal worden opgedeeld in 3 onderdelen: de proof-of-concept applicatie, de analyselibrary en de software die de library die de analyse zal verbeteren m.b.v. machine learning.

Dat de proof-of-concept applicatie losstaat is vanzelfsprekend. Deze applicatie moet apart ontwikkeld worden omdat deze alleen is bedoeld voor gebruik tijdens het ontwikkeltraject en/of demonstraties.

De machine learning library zal apart van de analyse library worden ontwikkeld omdat de analyse library de machine learning niet nodig heeft om te kunnen werken. De machine learning wordt alleen gebruikt om de analyse te verbeteren. Aangezien dit een langzaam proces zal zijn, zal dit verbeteren niet gebeuren tijdens het gebruik.

Naast deze 3 onderdelen wordt er uiteraard gebruik gemaakt van de microfoon en/of het filesystem van het systeem waar het op draait.

Proof-of-concept applicatie

De proof-of-concept applicatie zal zo simpel als mogelijk zijn, zonder aan functionaliteit in te leveren. Een GUI, een Control-class en een adapter voor de analyselibrary. Er zullen 2 mogelijkheden zijn voor de input: microfoon of file.

Analyse library

De analyse library zal worden opgedeeld in de volgende packages:

- Control
- APIBundle
- Analysis

Control package

Vanuit de Control package zullen de andere packages worden aangestuurd. De Control class zal het aanspreekpunt van de library zijn voor toegang van buiten de library.

APIBundle package

Dit is de package waar de externe API's bij elkaar komen, gemanaged worden en waar de output van de externe API's gestandaardiseerd wordt. Dit gebeurt in deze package zodat er verder nergens rekening gehouden hoeft te worden met de verschillen tussen de API's of libraries.

ExternalSpeechAPIAdapterManager

Deze class is singleton en zal de ExternalSpeechAPIAdapters bijhouden. Ook zal deze een geschikte API of library kunnen selecteren op basis van criteria, bijvoorbeeld de ondersteuning van een taal of aspect.

ExternalSpeechAPIAdapters

De ExternalSpeechAPIAdapters zijn elk verantwoordelijk voor het afhandelen van de input vanuit de manager, de API-aanroep, de verwerking en standaardisering van wat de API terug geeft en de output naar de manager. Elke ExternalSpeechAPIAdapter is verantwoordelijk voor één API.

ExternalSpeechAPIResult

Het gestandaardiseerde resultaat van een externe API zal worden opgeslagen in een object van deze class.

AudioUtilities

Aangezien niet elke externe API of library hetzelfde audioformaat accepteert, zal in de meeste gevallen de bitrate, samplesize, mono/stereo of bestandsformaat moeten worden aangepast. De methodes om dit te doen worden gebundeld in deze class zodat ze hergebruikt kunnen worden voor meerdere externe API's of libraries.

Analysis package

In de Analysis package zal de daadwerkelijke analyse van de resultaten van de API's plaatsvinden. Er zal per aspect een analyse worden uitgevoerd en een resultaat worden gegenereerd.

SpeechAspect

SpeechAspect is de enum met alle beschikbare aspecten. Deze zal gebruikt worden om aan te geven welke aspecten door een externe API of library worden ondersteund, om aan te geven voor welk aspect een SpeechAspectAnalysisResult het resultaat is en om aan te geven bij welk aspect een bepaalde implementatie van *ISpeechAspectAnalysis* hoort.

Aan de hand van bij welk aspect een resultaat hoort, wordt in ResultAnalysis met de juiste data vergeleken.

AnalysisControl

De class AnalysisControl zal verantwoordelijk zijn voor het bijhouden van de SpeechAspectAnalysisManager en van alle resultaten die van de analyses. Ook zal deze klasse ResultAnalysis aansturen.

ISpeechAspectAnalysis

ISpeechAspectAnalysis is de interface die door elke analyse-class wordt geïmplementeerd. In AnalysisControl zal een lijst van ISpeechAspectAnalysis objecten worden bijgehouden.

SpeechAspectAnalysisManager

Deze class zal alle *ISpeechAspectAnalysis* objecten bijhouden en een *ISpeechAspectAnalysis* kunnen selecteren op basis van een SpeechAspect.

SpeechAspectAnalysisResult

In deze class zal het resultaat van een analyse worden opgeslagen. Objecten van deze class dienen als input voor de ResultAnalysis.

ResultAnalysis

In ResultAnalysis zal het SpeechAspectAnalysisResult worden vergeleken met de verzamelde data. AnalysisResult zal het resultaat hiervan aan alle AnalysisResultListeners doorgeven. Er is gekozen voor een constructie met listeners zodat er makkelijk aan meerdere objecten een resultaat kan worden terug gegeven. De AnalysisControl class is een AnalysisResultListener zodat in de AnalysisControl een array van resultaten of iets dergelijks kan worden bijgehouden.

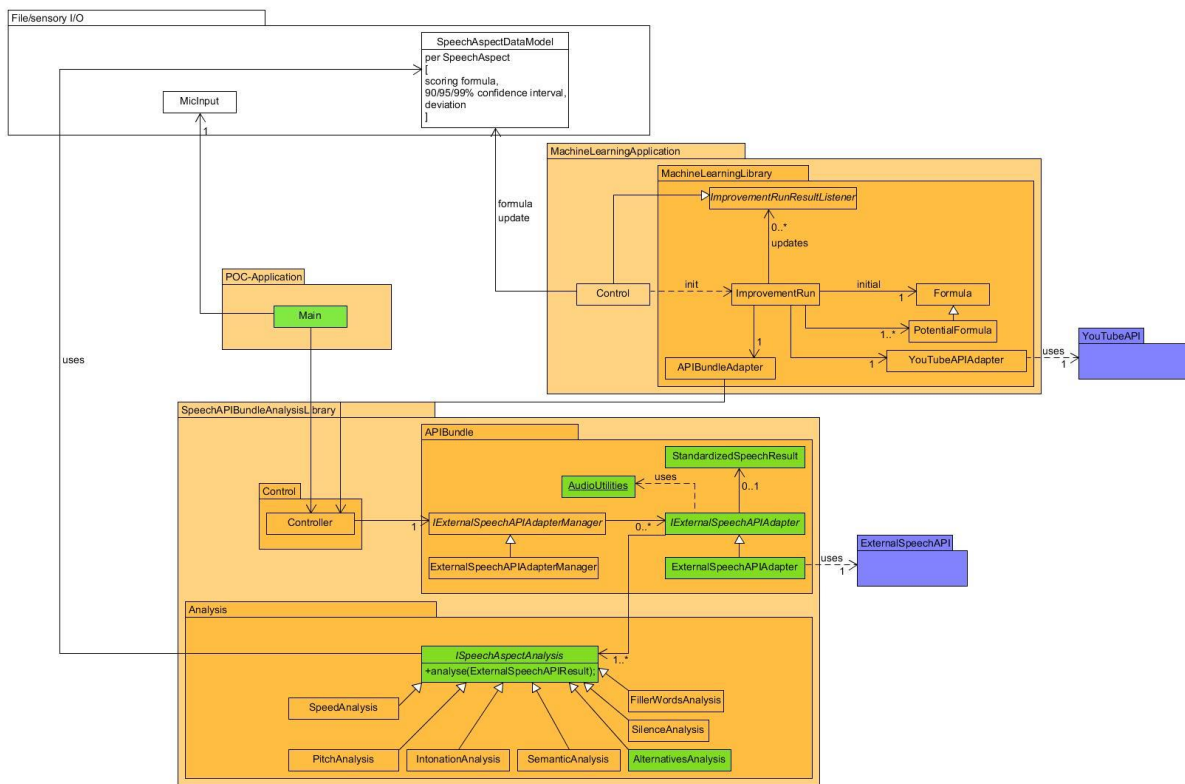
MachineLearning

Het machine learning onderdeel is voor een groot deel nog niet zeker. In grote lijnen zal er binnen één run (ImprovementRun) een formule (Formula) zijn die verbeterd moet worden. De mogelijk betere formules (PotentialFormula) worden getest en vergeleken met de te verbeteren formule. Dit zal gebeuren aan de hand van YouTube statistieken m.b.v. de YouTube API (YouTubeAPIAdapter) en de spraakanalyselibrary (APIBundleAdapter). Resultaten van zo'n run zullen worden doorgegeven aan alle objecten die dat willen (*ImprovementRunResultListener*) waaronder de applicatie die het geheel aanstuurt (Control).

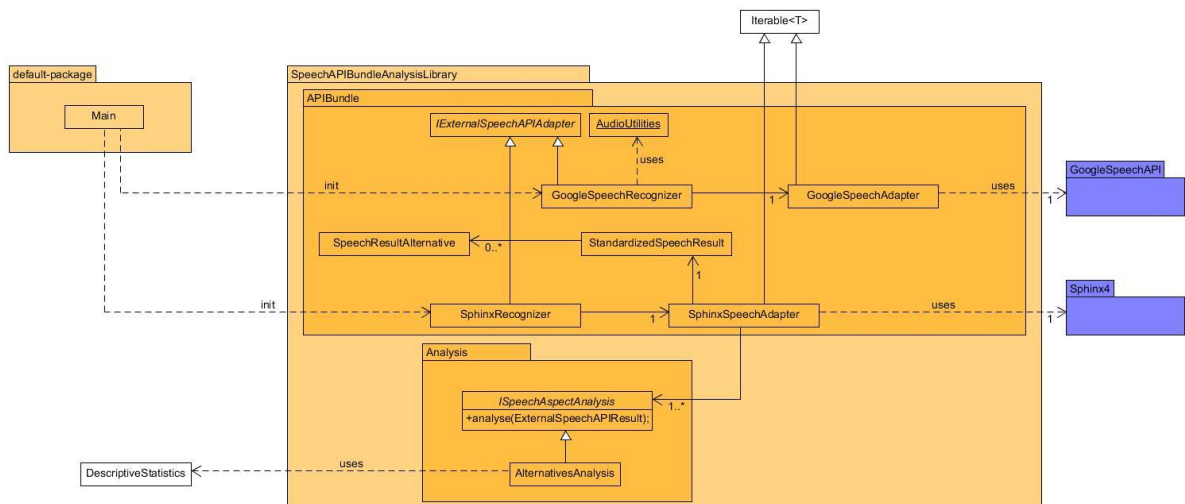
Het is waarschijnlijk dat er uiteindelijk een library gebruikt zal worden aangezien machine learning veel werk is om zelf in elkaar te zetten en een library waarschijnlijk accurater is.

Voortgang 12-06-2015

Bij deze eerste implementatie heb ik één aspect geïmplementeerd voor beide API's. De groengekleurde onderdelen zijn geïmplementeerd. Echter nog niet in de bedoelde structuur.

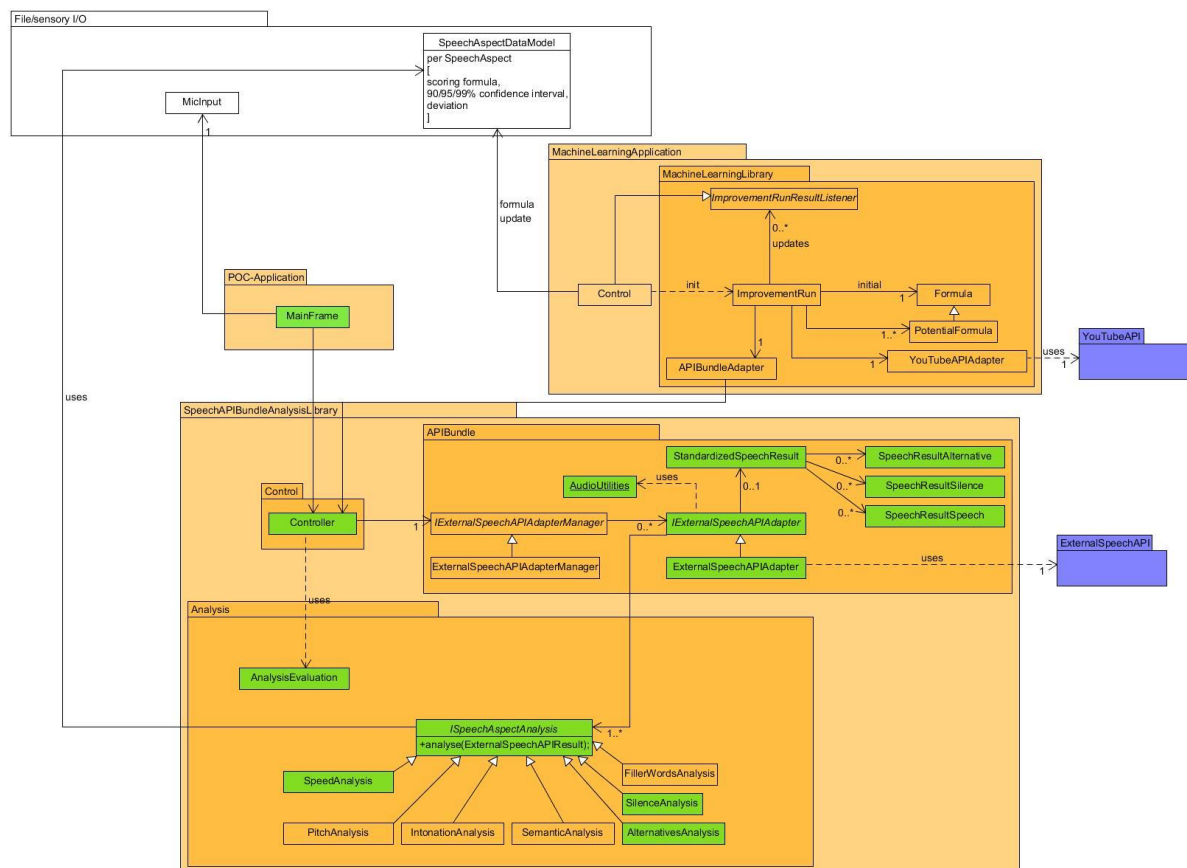


Het diagram van wat er tot nu toe is geïmplementeerd ziet er als volgt uit:



Beide recognizers hebben een `Iterable<T>` die de resultaten van de API ophaalt, maar die is niet een harde eis. Ik kan er naar mijn idee niet van uit gaan dat bij elke API een `Iterable<T>` kan worden toegepast.

Voortgang 24-07-2015

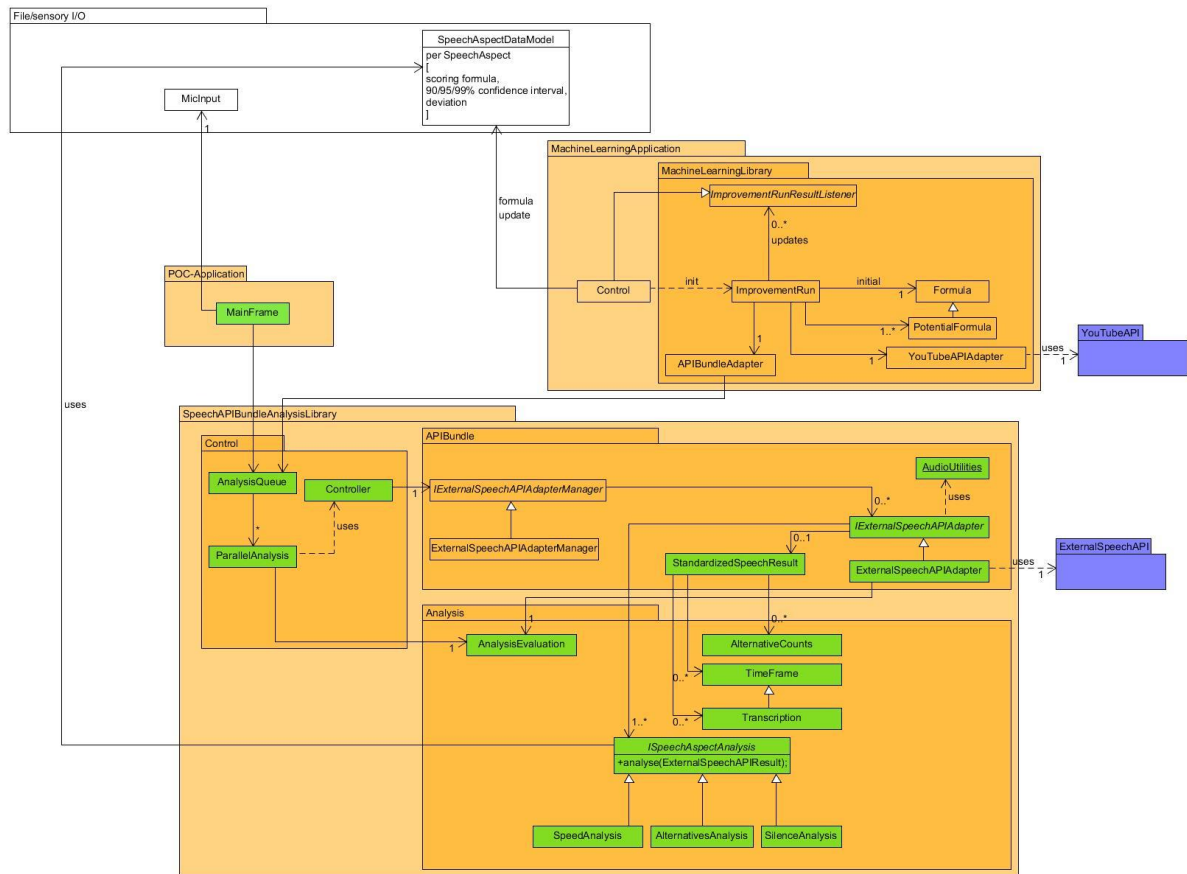


Bij deze iteratie heb ik nog twee analyses geïmplementeerd. Daarnaast bleek dat Google niet betrouwbaar resultaten geeft. Er komen soms wel en soms geen resultaten terug. Ik implementeer daarom voor het Proof of Concept alleen Sphinx. Sphinx is hier weergegeven als ExternalSpeechAPIAdapter.

Een ander nieuw onderdeel is de AnalysisEvaluation. Dit is de klasse die op basis van de gegeneraliseerde resultaten uit de API of API's een cijfer kan geven aan de presentatie.

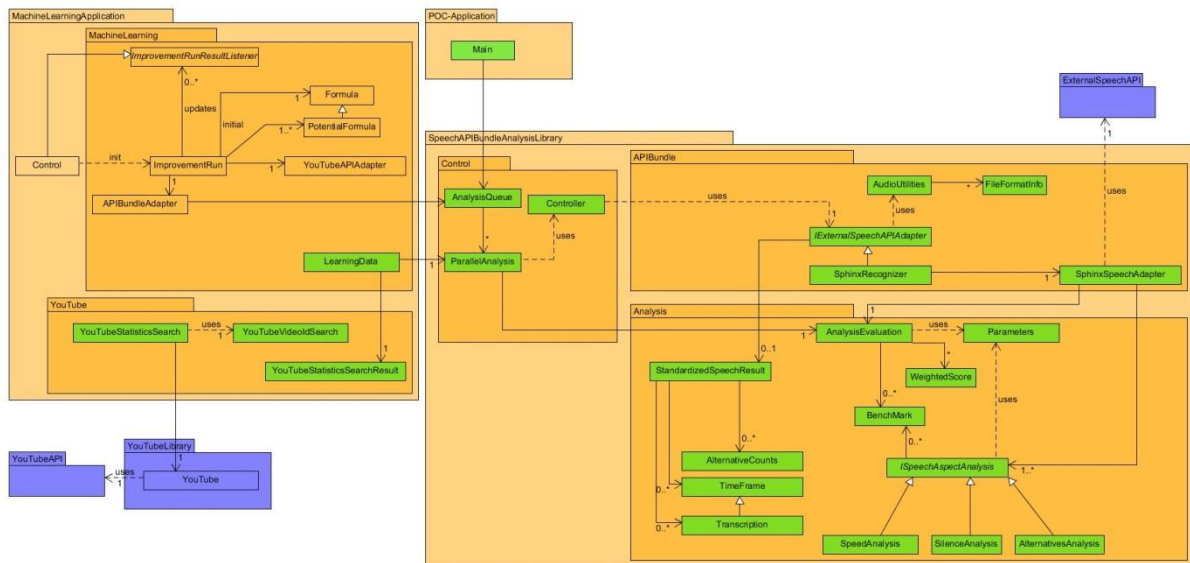
Ten slotte is de GUI er bij gekomen. Deze is vrij simpel en spreekt communiceert alleen via de Controller.

Voortgang 30-07-2015



Dit is de iteratie na de feedback van mijn begeleider over het refactoren. Veel benamingen zijn algemener, zoals `Transcription` i.p.v. `SpeechResultSpeech`. Ook heb ik `threading` geïmplementeerd. `ParallelAnalysis` is een `Runnable`. Er wordt een wachtrij van bestanden in `AnalysisQueue` bijgehouden. De `Controller` class is nog altijd verantwoordelijk voor het aanroepen van de `IExternalSpeechAPIAdapter`, in dit geval Sphinx. Het resultaat van zo'n aanroep is een `AnalysisEvaluation` object. Hierin worden verschillende cijfers en wegingen bijgehouden.

Voortgang 24-08-2015



Na nog een keer feedback te hebben gekregen over de refactoring van mijn code, heb ik nog een aantal aanpassingen gedaan:

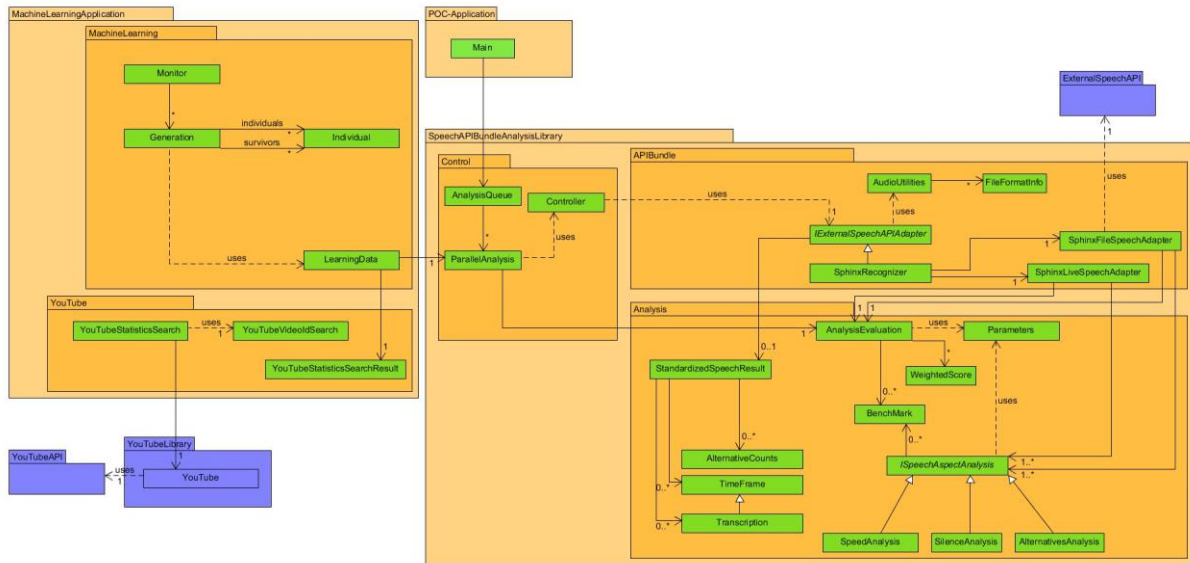
- De getallen die ogenschijnlijk bij elkaar horen, zoals een cijfer voor een goede en een zeer goede presentatie, worden nu bijgehouden in één class, de Benchmark.
- De scores, wegingen en de berekeningen die daarbij komen kijken horen ook bij elkaar. Deze heb ik samen in de WeightedScore class gezet.
- De StandardizedSpeechResult class en de bijhorende classes zijn verplaats naar de Analysis package. Naar mijn idee horen deze hier thuis omdat ze de basis vormen voor de analyses.

Daarnaast heb ik nog een aantal onderdelen toegevoegd. De YouTube-API kan nu gebruikt worden om de statistieken van een geanalyseerde presentatie op te zoeken. Deze statistieken worden opgeslagen in een `YouTubeStatisticsSearchResult` object. Dit object, samen met de `ParallelAnalysis` waar deze bij hoort, vormen de data voor de machine learning: `LearningData`. De onderdelen voor de afhandeling van de YouTube API aanroepen zitten in de `MachineLearning-package` omdat de analyse de YouTube API niet gebruikt. Er wordt alleen gebruik gemaakt van de YouTube API voor de machine learning.

Om de analyse library voor te bereiden op de toepassing van machine learning heb ik de class Parameters toegevoegd. Hierin worden alle variabele waarden die de machine learning zal afleiden en aanpassen opgeslagen. Op het moment van aanmaken bestaat de class alleen nog uit een HashMap waarin de parameters hard-coded zijn opgeslagen. Na het implementeren van de machine learning zullen de parameters waarschijnlijk uit een bestand worden geladen.

Voortgang 01-10-2015 – Afronding

Hieronder het uiteindelijke ontwerp van de gehele ontwikkelde library.



Wederom zijn de groengekleurde onderdelen de onderdelen die ik zelf geïmplementeerd heb en de blauwgekleurde onderdelen de externe API's.

In dit laatste ontwerp zijn de onderdelen voor de verwerking van live presentaties en de onderdelen voor de machine learning toegevoegd.

Bijlage G – Resultaten 10 bestanden

Hitchens

Alternatives score: 9.74452498405987
Silence score: 7.400461293031389
Speed score: 8.593726512000803
Total score: 8.428643329538247
Time taken: 85

Death to pennies

Alternatives score: 8.073408503641062
Silence score: 6.192250559702522
Speed score: 8.462719017777207
Total score: 7.18149334593881
Time taken: 84

Motivation

Alternatives score: 7.1494891146053945
Silence score: 6.039225429408094
Speed score: 9.916818212433489
Total score: 6.940273409235256
Time taken: 188

Starcraft

Alternatives score: 5.034790272215743
Silence score: 7.841916079196943
Speed score: 8.697025408743132
Total score: 6.896132567772266
Time taken: 1642

Bottom

Alternatives score: 6.922459872788319
Silence score: 6.336885812191349
Speed score: 7.874372393888024
Total score: 6.748661907627296
Time taken: 144

Breath holding

Alternatives score: 2.0844142230292504
Silence score: 6.807689038402648
Speed score: 5.603813974594885
Total score: 4.885976599661654
Time taken: 273

Douglas Adams- Parrots the Universe and Everything

Alternatives score: 4.534103260625093
Silence score: 1.1635914172605903
Speed score: 6.93938572261634
Total score: 3.1495076466917475
Time taken: 1456

The Shores of the Cosmic Ocean

Alternatives score: 4.704758204758204
Silence score: -6.031761490370725
Speed score: 3.7995434860334676
Total score: -0.7766534826468523
Is this a presentation?
Time taken: 446

Old spice

Alternatives score: 5.196345514950165
Silence score: -29.23595920783131
Speed score: 3.592637479169823
Total score: -14.708358331750727
Is this a presentation?
Time taken: 8

Pitch perfect

Alternatives score: 7.586378737541529
Silence score: -166.13311441398545
Speed score: 1.2643911511611154
Total score: -92.58504414709965
Is this a presentation?
Time taken: 17

Bijlage H – Requirements

Hieronder de requirements voor de ontwikkelde library. De requirements komen hoofdzakelijk van de heer van Tilborg. Daarnaast zijn er ook requirements die ik in overleg met mijn begeleider heb toegevoegd.

Requirement	Prioriteit
De library moet geschreven worden in Java	Must
De library moet Nederlands ondersteunen	Should
Verstaanbaarheid analyseren	Should
Snelheid analyseren	Could
Stiltes analyseren	Could
Aantal keer 'eh' tellen	Could
Verloop toonhoogte analyseren	Could
Ondersteuning voor live presentaties	Must
Export van resultaten naar CSV	Must
Resultaten analyse verbeteren met machine learning	Must
Meerdere API's kunnen ondersteunen	Must
Alle classes en methoden die nodig zijn om het product te gebruiken documenteren	Must

Bijlage I – API-documentatie

Hieronder alle classes, public methoden en public velden die nodig zijn voor gebruik van de library.

- Class: **AnalysisQueue**. Hierin wordt een wachtrij aan bestanden bijgehouden die nog geanalyseerd moeten worden.
 - o Constructor: default.
 - o Veld: `queuedAnalyses`. Dit is een `ArrayList<String>` met alle locaties van bestanden die nog geanalyseerd moeten worden erin.
 - o Veld: `finishedAnalyses`. Dit is een `ArrayList<ParallelAnalysis>` met alle analyses van de verwerkte bestanden erin.
 - o Methode: `add(String)`. Voegt de bestandslocatie van een audiobestand toe aan `queuedAnalyses`.
- Class: **ParallelAnalysis**. Extends `Runnable`. Deze class representeert de analyse van één bestand. Deze class wordt hoofdzakelijk door `AnalysisQueue` gebruikt.
 - o Constructor: `ParallelAnalysis(String)`. Initieert het `fileLocation` veld met de meegegeven waarde.
 - o Veld: `fileLocation`. De bestandslocatie van het te analyseren of geanalyseerde bestand.
 - o Methode: `run()`. Hiermee kan de analyse op een aparte Thread gedaan worden.
 - o Methode: `getEvaluation()` en `setEvaluation(AnalysisEvaluation)`. Methodes voor het verkrijgen en aanpassen van de private `AnalysisEvaluation`.
 - o Methode: `toString()`. Retournt de String-representatie van de `ParallelAnalysis`.
 - o Methode: `writeAnalysisToFile(String)`. Schrijft de analyse weg naar het bestand op de meegegeven plaats.
- Class: **Controller**. De class tussen de `AnalysisQueue` en de spraakherkenningslibraries. Alle methoden in deze class zijn static.
 - o Constructor: default. Niet nodig omdat alle methoden static zijn.
 - o Methode: `analyzeAudioFile(String)`. Analyseert het bestand op de meegegeven bestandslocatie.
 - o Methode: `toggleLive()`. Initialiseert, pauzeert of hervat de live presentatieanalyse.

Bijlage J – Afstudeerplan

Afstudeerplan

Informatie afstudeerder en gastbedrijf

Afstudeerblok: 2015-1.2 (start uiterlijk 11 mei 2015)
Startdatum uitvoering afstudeeropdracht: 1 mei 2015
Inleverdatum afstudeerdossier volgens jaarrooster: 5 oktober 2015

Studentnummer: 11098511
Achternaam: dhr de Haan
Voorletters: B
Roepnaam: Bas
Adres: Churchilllaan 83
Postcode: 2421GS
Woonplaats: Nieuwkoop
Telefoonnummer: -
Mobiel nummer: 06-33667474
Privé emailadres: basdehaan@live.com

Opleiding: Informatica
Locatie: Den Haag
Variant: voltijd

Naam studieloopbaanbegeleider: O. Zor
Naam begeleidend examiner: E.M. van Doorn
Naam tweede examiner: G.M. Tuk

Naam bedrijf: Ordina
Afdeling bedrijf: J-technologies, de Java afdeling
Bezoekadres bedrijf: Ringwade 1
Postcode bezoekadres: 3439LM
Postbusnummer: 7101
Postcode postbusnummer: 3430JC
Plaats: Nieuwegein
Telefoon bedrijf: 030 663 70 00
Telefax bedrijf:
Internetsite bedrijf: www.ordina.nl

Achternaam opdrachtgever: dhr van Tilborg
Voorletters opdrachtgever: R
Titulatuur opdrachtgever:
Functie opdrachtgever: Partner innovative technologies
Doorkiesnummer opdrachtgever:
Email opdrachtgever: richard.van.tilborg@ordina.nl

Achternaam bedrijfsmentor: dhr Verbruggen
Voorletters bedrijfsmentor: F
Titulatuur bedrijfsmentor:
Functie bedrijfsmentor: Software Architect
Doorkiesnummer bedrijfsmentor:
Email bedrijfsmentor: frank.verbruggen@ordina.nl

Doorkiesnummer afstudeerder:

Functie afstudeerder (deeltijd/duaal): Developer

Titel afstudeeropdracht: Het ontwikkelen van een spraakanalyselibrary bij Ordina.

Opdrachtomschrijving

Bedrijf

Ordina ontwikkelt business- en IT-oplossingen voor verschillende markten. Deze markten zijn de industrie, de financiële wereld, de overheid en de zorg.

Ordina is een bedrijf uit 1983, heeft zo'n 2.500 medewerkers en realiseert momenteel een lichte omzetgroei. De meeste omzet, 67%, komt van de Professional Services & Projects divisie. Hier is de J-technologies afdeling onderdeel van. Het primaire proces van de J-technologies afdeling is het ontwikkelen van Java-software. Ook worden trainingen georganiseerd door Ordina, waaronder presentatietrainingen. De opdracht heeft betrekking op de presentatietrainingen en wordt uitgevoerd binnen de J-technologies afdeling.

Probleemstelling

Binnen de afdeling van Ordina die de omscholing en trainingen verzorgt, worden presentatietrainingen gegeven. Er is bij deze trainingen behoefte aan software, die op basis van een spraakanalyse bepaalt hoe goed een presentatie is. Met zo'n analyse kan objectief feedback gegeven worden over hoe mensen beter hun stem kunnen gebruiken en uitspraak kunnen verbeteren.

Doelstelling van de afstudeeropdracht

De doelstelling van de opdracht is het ontwikkelen van een Java-library, die spraakanalyse uitvoert om feedback te geven over de mate van verstaanbaarheid en articulatie van een presentatie.

Om deze feedback te kunnen geven, zal een library gebruikt worden. Dit is niet een library die Ordina heeft klaarliggen, maar er zal een klein onderzoek uitgevoerd worden naar welke open-source libraries bruikbaar zijn. Een voorbeeld van zo'n open-source library is Praat. Deze kan onder andere de hoogte van het spreken, de energie waarmee gesproken wordt en de mate van articulatie bepalen.

Voor het achterhalen van de requirements zullen gesprekken met de presentatoren van Ordina gehouden worden.

Voor de validatie van de feedback zal een oplossing bedacht worden. Er kan bijvoorbeeld vergeleken worden met bekende mensen die veel presenteren of veel in het openbaar spreken zoals politici.

Resultaat

Als de opdracht succesvol is uitgevoerd, zullen de genoemde aspecten van een presentatie gemeten kunnen worden door de library. Er zal een uitspraak gedaan kunnen worden over de validiteit van deze metingen. Daarnaast zal een losstaande proof-of-concept applicatie geschreven zijn. De library kan gebruikt worden in andere applicaties op basis van de bijpassende API-documentatie.

Uit te voeren werkzaamheden, inclusief een globale fasering, mijlpalen en bijbehorende activiteiten

Binnen Ordina is Scrum de standaard voor projecten. Aangezien er niet in een team gewerkt zal worden, zullen aspecten van Scrum gebruikt worden. Er zal gewerkt worden in sprints van 2 weken, met een totaal van 10 sprints.

Er is binnen de J-technologies afdeling van Ordina een standaard set aan tools. Deze tools zijn o.a. de IDE Eclipse, het Java-framework Spring en het document management systeem LinkedData Virtuoso. Aan het begin van het project zal worden bepaald of het Spring framework nuttig is voor dit project. LinkedData Virtuoso wordt gebruikt om binnen Ordina documenten bij te houden en te delen met andere werknemers.

Er wordt voor de mijlpaalproducten uitgegaan van een totaal van 70 dagen. De volgende planning geeft een indruk van de verwachte tijdsbesteding per onderdeel.

Globale planning:

Start:

- Plan van aanpak schrijven
2 dagen
- Kort onderzoek open-source libraries
4 dagen

Iteratief:

Opstellen, prioriteren en overeenstemmen requirements. Dit aan de hand van interviews/gesprekken met medewerkers van Ordina. De requirements worden ingedeeld aan de hand van het MoSCoW-principe.

- 7 dagen
- Literatuuronderzoek naar spraakanalyse
5 dagen
- Ontwerp van het product. Dit zal gedaan worden m.b.v. UML in zowel analyse- als design diagrammen
10 dagen
- Bouw van het product
15 dagen
- Refactoren van de code
2 dagen
- Testen. De library zal aan de hand van een testplan per module en vervolgens ook in zijn geheel getest worden. Hiervoor zijn audiobestanden beschikbaar op internet.
8 dagen

Afronding:

- Implementeren en overdracht. Hier valt het schrijven van de API-documentatie onder.
2 dagen
- Schrijven van afstudeerdossier
15 dagen

Op te leveren (tussen)producten

De volgende producten zullen tijdens het project worden opgeleverd:

- Plan van aanpak
- Onderzoeksrapport over de librarykeuze
- Lijst met requirements
- Ontwerp van het product
- Voortgangsrapport na elke sprint
- Testplan
- Testrapport
- Spraakanalyselibrary
- API-documentatie
- Proof-of-concept applicatie

Te demonstreren competenties en wijze waarop

Selecteren methoden, technieken en tools - niveau 3

De te ontwikkelen software zal gebruik maken van een bestaande spraakherkenningslibrary. Deze library zal aan de hand van een kort onderzoek gekozen worden. Ook zal bepaald worden of het Spring framework nuttig is voor dit project.

Ontwerpen systeemdeel - niveau 4

De te ontwikkelen library zal ontworpen worden als herbruikbaar onderdeel van een systeem. De grote lijnen van de structuur van de library zullen bepaald worden, zodat deze herbruikbaar is. De library zal ontworpen worden m.b.v. UML.

Bouwen applicatie - niveau 4

Ik zal de library bouwen en om te kunnen laten zien dat de library werkt, zal een proof-of-concept applicatie gebouwd worden. Deze applicatie zal gebruik maken van de library, maar er verder los van staan. Zowel de library als de applicatie zullen gebouwd worden in Java.

Initiëren en plannen van het testproces - niveau 3

Aan het eind van elke sprint zal de ontwikkelde software getest worden. Daarbij zullen verschillende testsoorten toegepast worden afhankelijk van wat getest moet worden. Dit zal gedocumenteerd worden in een testplan.

Uitvoeren en rapporteren van het testproces - niveau 3

Als afsluiting van een sprint zullen de tests uitgevoerd, gedocumenteerd en gerapporteerd worden.

Bijlage K – Tussentijdse beoordelingen

Bespreking concept	Tussentijds assessment	Eerste beoordeling
--------------------	------------------------	--------------------

Formulier bespreking concept afstudeerdossier

Student: Bas de Haan

Studentnummer: 11098511

Datum: 31 augustus 2015

Tijdens de bespreking is het volgende geconstateerd:		ja	nee
a	Het voortgangsverslag is ontvangen	X	
b	Het afstudeerdossier is digitaal beschikbaar	X	
c	Het afstudeerdossier is opgebouwd conform de richtlijnen	X	
d	Het goedgekeurde afstudeerplan is aanwezig	X	
e	Het plan van aanpak is aanwezig	X	
f	Reeds geleverd commentaar is aanwezig	X	
g	Het afstudeerdossier geeft voldoende inzicht in de stand van zaken	X	
h	De afstudeeropdracht is tot nu toe naar behoren uitgevoerd	X	

Verbeterpunten:

Probleemstelling en Plan van aanpak beschrijven. Besteed aandacht aan de planning, methoden, technieken, risico's en tegenmaatregelen.

Beschrijf de problemen, oplossingsalternatieven en de gemotiveerde keuze die je gemaakt hebt.

Neem zinvolle delen van je producten inclusief toelichting op in je eindverslag. Zo ontbreken classdiagrammen.

Opmerkingen:

Je hebt al veel beschreven.

Naam begeleidend examiner: E.M. van Doorn

Datum: 31 augustus 2015

Bespreking concept	Tussentijds assessment	Eerste beoordeling
--------------------	------------------------	--------------------

Formulier tussentijds assessment

Student: Bas de Haan

Studentnummer: 11098511

Datum: 11 september 2015

eerste / ~~tweede~~ TTA:

Tijdens het tussentijds assessment is het volgende geconstateerd:		ja	nee
a	Het voortgangsverslag is ontvangen	X	
b	Het afstudeerdossier is digitaal beschikbaar	X	
c	Het afstudeerdossier is opgebouwd conform de richtlijnen	X	
d	Het goedgekeurde afstudeerplan is aanwezig	X	
e	Het plan van aanpak is aanwezig	X	
f	Reeds geleverd commentaar is aanwezig	X	
g	Het afstudeerdossier geeft voldoende inzicht in de stand van zaken	X	
h	De afstudeeropdracht is tot nu toe naar behoren uitgevoerd	X	

Aanpak	O	T	V	G
Passend		X		
Theoretisch verantwoord		X		
Samenhang uitvoering beroepstaken			X	

Beroepstaken op afgesproken niveau uitgevoerd?		O	T	V	G
1	Selecteren methoden, technieken en tools - niveau 3		X		
2	Ontwerpen systeemdeel - niveau 4			X	
3	Bouwen applicatie - niveau 4			X	
4	Initiëren en plannen van het testproces - niveau 3		X		
5	Uitvoeren en rapporteren van het testproces - niveau 3		X		

6					
Producten		O	T	V	G
<i>Tussenproducten</i>				X	
<i>Eindproducten</i>				X	

Effectief communiceren	O	T	V	G
<i>Binnen afstudeerbedrijf</i>			X	
<i>Afstudeerdossier</i>			X	

Reflectie	O	T	V	G
<i>Inzicht in eigen functioneren</i>		X		
<i>Inzicht in eigen leerproces</i>		X		

Toelichting per beoordelingscriterium

Aanpak
Er wordt in het verslag gezegd dat er delen van Scrum worden gebruikt, maar in feite is dat alleen maar het werken in sprints. Niet duidelijk is waarom die telkens twee weken moeten zijn.

Beroepstaken op afgesproken niveau uitgevoerd?
De theoretische onderbouwing (statistiek) en conclusies laten te wensen over. Het is de vraag in hoeverre de student dat is aan te rekenen. In feite is hij verantwoordelijk voor het ontwikkelen van de tools zelf.

Producten

Ten tijde van het TTA was het nog niet goed te beoordelen of de gemaakte producten naar tevredenheid functioneren. In de week na het TTA zal dat pas duidelijk worden.

Effectief communiceren

Het verslag is in taalgebruik acceptabel maar bevat te veel onnodige informatie en is daardoor onnodig dik. Redeneringen zijn niet altijd goed te volgen.

Reflectie

Reflectie nog ontbreekt vrijwel geheel in het verslag.

Advies

Aankruisen welk advies de student heeft gekregen

<input checked="" type="checkbox"/>	Inleveren (bindend advies)
<input type="checkbox"/>	Verlengen (vrijblijvend advies)
<input type="checkbox"/>	Stoppen (vrijblijvend advies)

Besluit student

Aankruisen welke beslissing de student heeft genomen (alleen na vrijblijvend advies)

<input type="checkbox"/>	Afstudeerdossier wordt op afgesproken datum ingeleverd	Inleverdatum: 5 oktober 2015
<input type="checkbox"/>	Afstudeerperiode wordt verlengd	Inleverdatum:
<input type="checkbox"/>	Student stopt met afstudeeropdracht	

Naam begeleidend examiner: E.M. van Doorn

Naam tweede examiner: G.M. Tuk

Datum: 9 september 2015

Bijlage L – Resultaten machine learning

De naamgeving is wat onduidelijk, maar het gaat hier om de verandering in de waarden.

Oorspronkelijke parameters

SPHINX_BENCHMARK_LOW_WORD_COUNT_GREAT: 7.0
SPHINX_BENCHMARK_LOW_WORD_COUNT_GOOD: 10.0
SPHINX_BENCHMARK_MED_WORD_COUNT_GREAT: 10.0
SPHINX_BENCHMARK_MED_WORD_COUNT_GOOD: 15.0
SPHINX_BENCHMARK_TOTAL_WORD_COUNT_GREAT: 7.5
SPHINX_BENCHMARK_TOTAL_WORD_COUNT_GOOD: 15.0
ALTERNATIVES_ANALYSIS_MAX_WORD_COUNT: 20.0
ALTERNATIVES_ANALYSIS_MAX_ALTERNATIVES: 1000.0
ALTERNATIVES_ANALYSIS_SPLIT_WORD_COUNT_LIMIT: 10.0
SILENCE_ANALYSIS_RATIO_WEIGHT: 1.0
SILENCE_ANALYSIS_SILENCE_WEIGHT: 2.0
SILENCE_ANALYSIS_SPEECH_WEIGHT: 1.0
SILENCE_ANALYSIS_GRADE_PER_POINT: 2.5
SILENCE_ANALYSIS_IDEAL_SILENCE_LENGTH: 481.77718832891232
SILENCE_ANALYSIS_IDEAL_SPEECH_LENGTH: 1027.26790450928385
SILENCE_ANALYSIS_IDEAL_SILENCE_SPEECH_RATIO: 1.2083490845383715
SILENCE_ANALYSIS_BENCHMARK_RATIO_STD_DEV_GREAT: 1.5726159614137725
SILENCE_ANALYSIS_BENCHMARK_RATIO_STD_DEV_GOOD: 6.365985598747253
SPEED_ANALYSIS_GUIDELINE: 130.0
SPEED_ANALYSIS_TOO_FAST: 200.0
SPEED_ANALYSIS_GRADE_ON_GUIDELINE: 10.0
SPEED_ANALYSIS_GRADE_TOO_FAST: 3.0
ANALYSIS_EVALUATION_SAMPLE_GRADE_GREAT: 8.1
ANALYSIS_EVALUATION_SAMPLE_GRADE_GOOD: 6.5
ANALYSIS_EVALUATION_ALTERNATIVES_LOW_COUNT_SCORE_WEIGHT: 0.5
ANALYSIS_EVALUATION_ALTERNATIVES_MED_COUNT_SCORE_WEIGHT: 0.25
ANALYSIS_EVALUATION_ALTERNATIVES_TOTAL_SCORE_WEIGHT: 0.0
ANALYSIS_EVALUATION_SILENCE_RATIOS_SCORE_WEIGHT: 1.2
ANALYSIS_EVALUATION_SILENCE_STD_DEV_SCORE_WEIGHT: 0.8
ANALYSIS_EVALUATION_SPEED_SCORE_WEIGHT: 1.0

Nieuwe parameters

SPHINX_BENCHMARK_LOW_WORD_COUNT_GREAT: 7.188102363670783
SPHINX_BENCHMARK_LOW_WORD_COUNT_GOOD: 8.763107139782287
SPHINX_BENCHMARK_MED_WORD_COUNT_GREAT: 8.602520364761164
SPHINX_BENCHMARK_MED_WORD_COUNT_GOOD: 16.87976411105433
SPHINX_BENCHMARK_TOTAL_WORD_COUNT_GREAT: 7.089096767953491
SPHINX_BENCHMARK_TOTAL_WORD_COUNT_GOOD: 14.252901393919739
ALTERNATIVES_ANALYSIS_MAX_WORD_COUNT: 23.05352310313124
ALTERNATIVES_ANALYSIS_MAX_ALTERNATIVES: 887.9443225935721
ALTERNATIVES_ANALYSIS_SPLIT_WORD_COUNT_LIMIT: 9.140604846113227
SILENCE_ANALYSIS_RATIO_WEIGHT: 1.0245145866763319
SILENCE_ANALYSIS_SILENCE_WEIGHT: 2.1279836542131103
SILENCE_ANALYSIS_SPEECH_WEIGHT: 1.1327006790198024
SILENCE_ANALYSIS_GRADE_PER_POINT: 2.02945103037864
SILENCE_ANALYSIS_IDEAL_SILENCE_LENGTH: 515.6230371391625
SILENCE_ANALYSIS_IDEAL_SPEECH_LENGTH: 895.6258065381016
SILENCE_ANALYSIS_IDEAL_SILENCE_SPEECH_RATIO: 1.320951344448337
SILENCE_ANALYSIS_BENCHMARK_RATIO_STD_DEV_GREAT: 1.6276445683484324

SILENCE_ANALYSIS_BENCHMARK_RATIO_STD_DEV_GOOD: 6.703656740615934
SPEED_ANALYSIS_GUIDELINE: 118.1371559594772
SPEED_ANALYSIS_TOO_FAST: 206.8894574108684
SPEED_ANALYSIS_GRADE_ON_GUIDELINE: 11.146531242661057
SPEED_ANALYSIS_GRADE_TOO_FAST: 3.4083246727331953
ANALYSIS_EVALUATION_SAMPLE_GRADE_GREAT: 8.458494981525726
ANALYSIS_EVALUATION_SAMPLE_GRADE_GOOD: 6.598805718272272
ANALYSIS_EVALUATION_ALTERNATIVES_LOW_COUNT_SCORE_WEIGHT: 0.5426793841472475
ANALYSIS_EVALUATION_ALTERNATIVES_MED_COUNT_SCORE_WEIGHT: 0.3043540511621277
ANALYSIS_EVALUATION_ALTERNATIVES_TOTAL_SCORE_WEIGHT: 0.04855967248910653
ANALYSIS_EVALUATION_SILENCE_RATIOS_SCORE_WEIGHT: 1.0563386395334022
ANALYSIS_EVALUATION_SILENCE_STD_DEV_SCORE_WEIGHT: 0.7900140805204892
ANALYSIS_EVALUATION_SPEED_SCORE_WEIGHT: 0.8913874273774938