TECHNISCHE HOCHSCHULE NÜRNBERG
GEORG SIMON OHM

Fakultät Informatik

Diarization und Overlap Detection für Arzt-Patienten-Gespräche

Bachelorarbeit im Studiengang Medieninformatik

vorgelegt von

Sebastian Klinger

Matrikelnummer 273 6425

am 31. Januar 2022

Erstgutachter:	Prof. Dr. Korbinian Riedhammer
Zweitgutachter:	Prof. Dr. Tobias Bocklet

TECHNISCHE HOCHSCHULE NÜRNBERG
GEORG SIMON OHM

Prüfungsrechtliche Erklärung der/des Studierenden

Angaben des bzw. der Studierenden:

Name: Klinger Vorname: Sebastian Matrikel-Nr.: 2736425

Fakultät: Informatik Studiengang: Medieninformatik

Semester: Wintersemester 21/22

Titel der Abschlussarbeit:
Diarization und Overlap Detection für Arzt-Patienten-Gespräche

Ich versichere, dass ich die Arbeit selbständig verfasst, nicht anderweitig für Prüfungszwecke vorgelegt, alle benutzten Quellen und Hilfsmittel angegeben sowie wörtliche und sinngemäße Zitate als solche gekennzeichnet habe.

Nürnberg, 31.01.22, S. Klinger
Ort, Datum, Unterschrift Studierende/Studierender

Erklärung zur Veröffentlichung der vorstehend bezeichneten Abschlussarbeit

Die Entscheidung über die vollständige oder auszugsweise Veröffentlichung der Abschlussarbeit liegt grundsätzlich erst einmal allein in der Zuständigkeit der/des studentischen Verfasserin/Verfassers. Nach dem Urheberrechtsgesetz (UrhG) erwirbt die Verfasserin/der Verfasser einer Abschlussarbeit mit Anfertigung ihrer/seiner Arbeit das alleinige Urheberrecht und grundsätzlich auch die hieraus resultierenden Nutzungsrechte wie z.B. Erstveröffentlichung (§ 12 UrhG), Verbreitung (§ 17 UrhG), Vervielfältigung (§ 16 UrhG), Online-Nutzung usw., also alle Rechte, die die nicht-kommerzielle oder kommerzielle Verwertung betreffen.

Die Hochschule und deren Beschäftigte werden Abschlussarbeiten oder Teile davon nicht ohne Zustimmung der/des studentischen Verfasserin/Verfassers veröffentlichen, insbesondere nicht öffentlich zugänglich in die Bibliothek der Hochschule einstellen.

Hiermit ☐ genehmige ich, wenn und soweit keine entgegenstehenden Vereinbarungen mit Dritten getroffen worden sind,

☒ genehmige ich nicht,

dass die oben genannte Abschlussarbeit durch die Technische Hochschule Nürnberg Georg Simon Ohm, ggf. nach Ablauf einer mittels eines auf der Abschlussarbeit aufgebrachten Sperrvermerks kenntlich gemachten Sperrfrist

von Jahren (0 - 5 Jahren ab Datum der Abgabe der Arbeit),

der Öffentlichkeit zugänglich gemacht wird. Im Falle der Genehmigung erfolgt diese unwiderruflich; hierzu wird der Abschlussarbeit ein Exemplar im digitalisierten PDF-Format auf einem Datenträger beigefügt. Bestimmungen der jeweils geltenden Studien- und Prüfungsordnung über Art und Umfang der im Rahmen der Arbeit abzugebenden Exemplare und Materialien werden hierdurch nicht berührt.

Nürnberg, 31.01.22, S. Klinger
Ort, Datum, Unterschrift Studierende/Studierender

Datenschutz: Die Antragstellung ist regelmäßig mit der Speicherung und Verarbeitung der von Ihnen mitgeteilten Daten durch die Technische Hochschule Nürnberg Georg Simon Ohm verbunden. Weitere Informationen zum Umgang der Technischen Hochschule Nürnberg mit Ihren personenbezogenen Daten sind unter nachfolgendem Link abrufbar: **https://www.th-nuernberg.de/datenschutz/**

Kurzdarstellung

Diese Arbeit beschäftigt sich mit Speaker Diarization. Sie hat zum Ziel ein Diarization System vorzubereiten, das potenziell mit Arzt-Patient-Gesprächen zurechtkommen könnte. Dazu wird vorausgesetzt, dass das System dazu in der Lage ist, mit Speech Overlap und einer flexiblen Anzahl an Sprechenden umzugehen. Weiter ist der AMI Korpus zum Testen und eval2000 Korpus zum Evaluieren vorgegeben worden. Für die Auswahl eines geeigneten Testsystems werden die besten drei Systeme auf Track 2 der DIHARD Challenge III und ein Subsystem auf Qualitätsmerkmale wie beispielsweise Diarization Error Rate (DER), Komplexität oder Echtzeitfähigkeit analysiert. Das Subsystem Self-Attentive (SA) End-to-end Neural Network-based speaker diarization model (EEND) mit Encoder-Decoder-Attractor (EDA) wird als Testsystem genutzt (`https://github.com/hitachi-speech/EEND`). Als Baseline dient ein Kaldi Standard Rezept für den AMI-Korpus mit Overlap Detection (`kaldi/egs/ami/s5c`). Letzteres implementiert ein Diarization System mit folgender Pipeline: Genutzt werden MFCC, Oracle SAD, X-Vektor Exktraktion mit vortrainiertem DNN, PLDA, Spectral Clustering und abschließend wird ein TDNN LSTM für Overlap Detection trainiert. Die bestehenden Diarization Systeme werden auf die vorgegebenen Datensets angepasst. Das Ergebnis auf dem eval2000 Set für die Baseline ergibt 84,9 % DER, für das Testsystem 100 % DER. Es wurden Hypothesen zum schlechten Abschneiden beider Systeme aufgestellt. Beispielsweise wird vermutet, dass ein Datensatz-Handhabungsfehler vorliegt, allerdings konnten keine Beweise zur Untermauerung der Hypothesen gefunden werden.

Inhaltsverzeichnis

Kapitel 1.

Einführung

1.1. Problemstellung und Motivation

Mit der Entwicklung immer größerer und günstigerer Speicher [Komo 14] [Klei 17] und der weltweiten Ausbreitung von Video- und Audiodiensten, wie beispielsweise YouTube, TikTok und Spotify, die es eben auch Otto Normalbürgern erlauben, einfach und schnell eigenen Content herzustellen und hochzuladen, aber auch durch das Wachstum von traditionellen TV-Kanälen, erhöht sich die gesammelte und gespeicherte Datenmenge von Video- und Tonmaterial ständig. So werden beispielsweise alleine auf YouTube minütlich 500 Stunden Videos hochgeladen [Wojc 20], das entspricht *30 000 Jahre* Videomaterial jährlich. Die elektronische Verarbeitung dieser Daten ist aufwändiger als bei vergleichsweise einfach auswertbaren Textdokumenten. Es ist etwa nicht ohne Weiteres möglich diese Daten nach Phrasen oder Inhalten zu durchzusuchen, um beispielsweise Videos für die Suche zu indizieren oder bei Nutzungsbedingungs-Verstöße zu sperren.

Das Teilgebiet der Informatik und Computerlinguistik, das sich mit der Lösung dieses Problems befasst, heißt Automatic Speech Recognition (ASR). Dabei werden speech-to-text Systeme genutzt, um den Audioanteil dieser Daten zu transkribieren. Die daraus resultierenden Textdokumente können dann für weitergehende Analysen und elektronische Datenverarbeitung genutzt werden. Eine Schwachstelle dieser Herangehensweise ist allerdings, dass die Transkripte keine Informationen über den jeweiligen Sprecher beinhalten. Das Fehlen solcher Informationen erschwert die Verarbeitung und Nutzung der Transkripte, wenn zum Beispiel Sprachmuster bestimmter Personen zu untersuchen sind. Dieses Problem ist insbesondere dann von Relevanz, wenn sich Redeabschnitte mehrerer Sprecher überschneiden, da den transkribierten Texten keine Zuordnung zu den Sprechern gemacht wird. Es werden also Modelle benötigt, die Metadaten zum Audiomaterial erzeugen können. Diese Forschungsfelder sind unter anderem im Bereich der Speaker oder Speech Diarization von Interesse. Diese hat das Ziel, eine Tonspur in ihre hörbaren Sprecher zu segmentieren und wiederzuerkennen. Die Erkennung gleichzeitig auftretender Sprecher stellt als Overlap eine besondere Herausforderung dar.

1.2. Ziel der Arbeit und Abgrenzung

Der Fakultät Informatik liegen Arzt-Patienten-Gespräche vor, die im Rahmen einer Studie mit der Klinik Nürnberg aufgezeichnet wurden. Um diese Daten für mögliche andere Forschungsprojekte nutzbar zu machen, sollen die Audiofiles nach Sprecheridentitäten unüberwacht segmentiert und klassifiziert werden. Nutzbar wären diese verarbeiteten Daten beispielsweise für Forschungsarbeiten bezüglich automatischer Erkennung von physischen oder psychischen Krankheiten, wie zum Beispiel Depression. Um eine möglichst reibungslose Weiterverarbeitung des Tonmaterials gewährleisten zu können, gilt es, die Daten bestmöglich zu segmentieren, um Arzt- und Patiententonspur klar voneinander zu trennen. Das Problem der Segmentierung und Klassifizierung soll durch maschinelle Lernverfahren gelöst werden.

Wegen datenschutzrechtlicher Hürden kann in dieser Arbeit nicht mit den Patientendaten gearbeitet werden. Um dennoch Vorarbeit leisten zu können, soll das vorrangige Ziel dieser Arbeit die Lösung des Problems der automatischen Segmentierung in einem speziellen Gesprächskontext sein. Dieser Kontext ist die Tonaufnahme eines Gesprächs in einem Raum zwischen mindestens zwei Personen. Da nicht zwangsläufig gegeben ist, dass nur zwei Personen am Gespräch beteiligt sind, muss das Modell eine flexible Anzahl von Sprechenden ermöglichen. Dies soll annähernd die Situation zwischen Arzt und Patient und gegebenenfalls Angehörige darstellen. Die Lösung des Problems soll mit existierenden Methoden und Modellen erarbeitet werden. Diese Arbeit hat es nicht zum Ziel ein generell gutes Modell für alle Situationen zu erarbeiten.

Ein weiteres Ziel ist es, dass das Diarization System mit Overlap Speech zurechtkommt. Es soll in weiterführenden Arbeiten möglich sein, die Art des Overlaps zu bestimmen. So soll beispielsweise erkannt werden können, ob es sich bei einem Sprachfragment um bestätigende „Hmm“s und „Aha“s handelt oder Dialog stattgefunden hat. Diese Unterscheidung könnte wichtig sein, möchte man automatisiert Gesprächsfragmente mit bestimmten Qualitätsmerkmalen herausfiltern.

Die Qualität der Diarizationsysteme soll in Hinblick auf Komplexität, Echtzeitfähigkeit („Onlinefähigkeit“) und Fehlerrate evaluiert werden. Letztere wird primär mit Diarization Error Rate (DER) und ihre Bestandteile wiedergegeben.

Für die Experimente wurden zwei Korpora vorgegeben. Zum Trainieren der Modelle soll der Augmented Multi-Party Interaction (AMI) und zur Evaluierung der 2000 HUB5 English Evaluation Speech (eval2000) Korpus verwendet werden.

1.3. Aufbau der Arbeit

Nach der Einführung in die Thematik ASR und Diarization und der Zielsetzung der Arbeit werden in Kapitel 2 fachliche Grundlagen erläutert. Es wird auf maschinelles Lernen, Sprache, sprachliche Verarbeitung, Feature Extraction, Speaker Segmentation, Speech Activation Detection, Modelling und Metriken zur Evaluation eingegangen.

Kapitel 3 erläutert die Hintergründe, nach denen geeignete Diarization Systeme für die Bearbeitung der Zielsetzung ausgewählt werden. Die engere Auswahl wird kurz beschrieben und anschließend bewertet. Das Ergebnis des Kapitels sind zwei Diarization Systeme. Zum einen die Baseline, die auf bewährte Methoden setzt, und zum anderen ein Testsystem, das neuartige und aktuelle Methoden verwendet.

Kapitel 4 beschreibt das Diarization Testsystem aus Kapitel 3 genauer und erläutert zum Teil die mathematischen Grundlagen. Das Kapitel basiert vorwiegend auf [Fuji 19a].

Im vorletzten Kapitel 5 werden technische Grundlagen erläutert, die notwendig sind, um die experimentelle Umsetzung und Testung der Diarization Systeme zu verstehen. Zudem wird auf die verwendeten Datensätze eingegangen. Abschließend werden die Ergebnisse analysiert.

Das Kapitel 6 fasst die Arbeit kurz zusammen und beschreibt Ansätze zur Verbesserung und Erweiterung der getesteten Systeme. Abschließende Gedanken zum Projekt beenden dann das Kapitel.

Kapitel 2.

Grundlagen

Dieses Kapitel beschreibt Grundlagen, die für das Verständnis dieser Arbeit notwendig sind. Neben der Eingrenzung und Beschreibung von maschinellem Lernen wird die typische Methodik und Herangehensweise an ein Projekt im Bereich der Speaker Diarization und Overlap Detection erklärt. Anschließend wird auf grundlegende technischen Hintergründe eingegangen.

2.1. Maschinelles Lernen

Maschinelles Lernen oder Machine Learning (ML) ist ein breit gefächerter Bereich der Informatik. Abstrakt geht es darum, einem Computer beizubringen, wie man bestimmte Aufgaben „erlernt“, das heißt aus vorhandenen Daten Strukturen herleitet. Aufgabengebiete können zum Beispiel das Erkennen von handgeschriebenen Buchstaben, eine computerunterstützte medizinische Diagnose, das Erkennen von Gesichtern oder das Vorschlagen von Produkten, die einen Kunden interessieren könnten, sein [Mell 18]. Als kleinen beispielhaften Einblick sollen folgende medienwirksame oder anderweitig prominente Produkte genannt werden:

AlphaGo ist ein Computerprogramm, das eine Reihe von Lernmethoden verwendet, um im Spiel Go professionelle Spieler zu besiegen. 2016 gelang es dem Programm Lee Sedol zu besiegen, der als weltbester Go-Spieler bekannt war. Go gilt als komplexer als Schach. Aufgrund eines deutlich größeren Spielbretts von 19×19 erhöht sich der theoretische Suchraum zum Ausprobieren aller Züge um googol[1] im Vergleich zu Schach [Silv 16].

Amazon bietet eine ML-Service-Komplettlösung an, mit der ein Kunde Vorschläge für seine Endkunden personalisieren kann. Hierbei wird ein variabler Aktivitäts-Stream wie zum Beispiel Klicks, Seitenaufrufe, Registrierungen, getätigte Einkäufe oder gesehene Medien benutzt, um personalisierte Produktvorschläge oder interessante Filme zu berechnen [Amaz].

[1] Eine googol entspricht $1,0 \times 10^{100}$

Ein weiteres und womöglich das bekannteste Beispiel ist die Internet-Suchmaschine Google, die Text-spezifische Lernverfahren verwendet, um die Antworten und ihr Ranking auf Suchanfragen zu optimieren.

2.2. Sprache

Sprache beim Menschen entsteht durch einen Luftstrom, der durch Druckaufbau in der Lunge erzeugt wird. Dieser Luftstrom durchquert die Luftröhre und passiert auf seinem Weg den Kehlkopf mit Stimmlippen, anschließend Rachen, Zunge, Mundhöhle, Zähne und Nasenraum und verlässt schlussendlich den Körper an den Lippen und Nase. Sind die Stimmlippen nah einander, werden sie in Schwingung versetzt und es entsteht ein Laut. Variieren lässt sich dieser Laut durch die Entfernung und Anspannung der Stimmlippen, durch Steuerung des Ausstoßes von Luft durch Mund oder Nase und durch filigrane Bewegungen der Zunge, Lippen und Mundmuskulatur. Durch dieses komplexe und koordinierte Zusammenspiel entstehen einzelne Laute, die aneinander gereiht Wörter, Sätze und damit Sprache ergeben [Eule 06, S. 5-6].

Die Variationen in den Luftstromänderungen erzeugen minimale Schwingungen, die durch die Luft propagiert werden. Diese Schwingungen sind Luftdruckunterschiede, die Schallwellen genannt werden. Diese Schallwellen werden im Ohr vom Trommelfell aufgenommen und über das Gehörknöchelchen auf mechanische Weise zum ovalen Fenster weitergeleitet. Am ovalen Fenster werden diese mechanischen Schwingungen an die Flüssigkeit in der Gehörschnecke hydraulisch weitergegeben. In der Gehörschnecke verteilt befinden sich Haarzellen, die je nach Region unterschiedlich empfindlich durch bestimmte Frequenzbereiche zu schwingen beginnen und Nervenimpulse auslösen, die entlang des Hörnervs in mehrere Hirnabschnitte der auditiven Wahrnehmung und Verarbeitung gelangen [Stad 09, S. 32-43].

Allein durch die Physiologie des Gehörs liegt die Vermutung nahe, dass die psychoakustische Wahrnehmung von Geräuschen nicht zwangsläufig linear erfolgen muss. Empirische Ergebnisse [Stev 37] zeigen, dass etwa ein Intervall zwischen 100 und 200 Hz als „Verdopplung“ der Tonhöhe wahrgenommen wird. Der Tonabstand zwischen 1,5 kHz und 1,6 kHz hingegen wird als deutlich geringer wahrgenommen. Die vermutete psychoakkustische Verdopplung bei 3 kHz stellt sich erst bei 6,3 kHz ein.

Ebenso wie die wahrgenommene Tonhöhe ist auch die Wahrnehmung der Lautstärke nicht-linear. Diese ist abhängig vom Frequenzbereich und wird in der Abbildung 2.1 veranschaulicht. Aus ihr lässt sich ablesen, dass in den hörbaren Randbereichen die Empfindlichkeit für die Wahrnehmung deutlich niedriger ist als im empfindlichen Kernbereich zwischen 0,5 - 5 kHz, in dem sich auch ein Großteil gesprochener Sprache befindet [Henn 08].

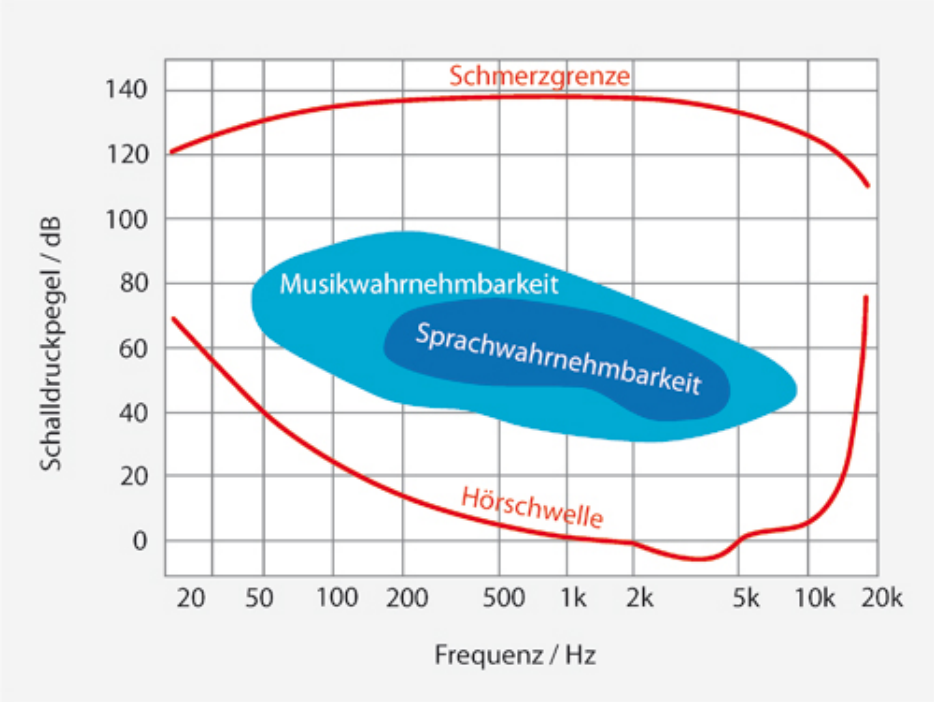

Abbildung 2.1.: Hörfläche begrenzt durch Hörschwelle und Schmerzgrenze [Bru14]

.

2.2.1. Diarization

Wie eingangs in Kapitel 1 angedeutet, befasst sich der Bereich der ASR mit mehreren Teilaufgaben:

- Spracherkennung
- Sprachsynthese
- Sprecheridentifikation
- Sprachsignalcodierung
- Sprachenidentifikation
- Sprechertransformation

Diarization umfasst Prozesse, um ein Audiosignal bezüglich der **Sprecheridentitäten** zu segmentieren. Es gilt also Personen auf Grundlage ihrer Äußerungen im Audiosignal zu erkennen [Pfis 08, S. 26-29]. Wie in Abbildung 2.2 angedeutet, wird von einem Diarization-System eine Audioquelle (Mix) in ihrer einzelnen Sprecher (A, B, C) segmentiert und zeitlich aufgelöst.

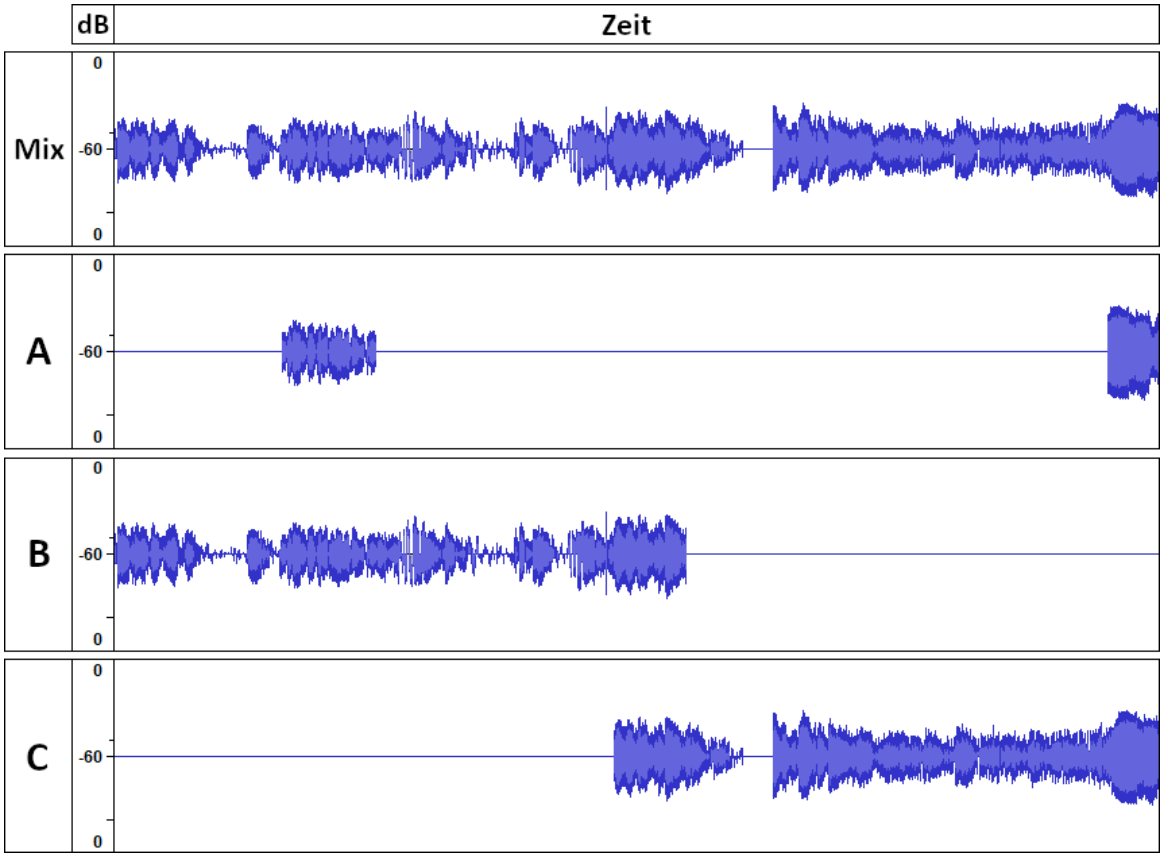

Abbildung 2.2.: Segmentiertes Audiosignal mit drei Sprechern.

2.2.2. Sprachverarbeitung

Auf Basis der erläuterten Besonderheiten der Sprachwahrnehmung ergibt sich, dass die primären Ansatzpunkte zur Verarbeitung auditiver Daten auf Grundlage der in Sprache vorhandenen Frequenzen f, Amplituden und in der Struktur der zeitlichen Abfolge dieser Daten liegen.

Mittels eines Mikrofons kann das analoge Schallsignal in ein digitales, also diskretes Signal umgewandelt werden. Die mindestens erforderliche Abtastrate, auch Samplingrate genannt, wird hierbei durch das Nyquist-Shannon-Abtasttheorem festgelegt.

$$f_{abtast} \geq \alpha \times f_{max} \mid \alpha \geq 2 \tag{2.1}$$

α wird in der Literatur vereinfacht mit 2 angegeben. Kleinere Abtastraten führen zu potenziellen Mehrdeutigkeiten und erzeugen sogenannte Aliasing-Artefakte, die wahrnehmbar die Aufnahme verschlechtern. Um beispielsweise 20 kHz aufzunehmen, sind mindestens 40 kHz Abtastfrequenz notwendig. Vor einer Abtastung muss ein Tiefpassfilter angewendet werden, um potenzielle Frequenzen über $f_{Abtast}/2$ herauszufiltern. Da durch Filter ebenfalls ungewollte Störeffekte auftreten können, wird α signifikant größer gewählt, damit ein Tiefpassfilter

nicht ungewollt Frequenzen unterhalb von 20 kHz filtert. Beispielsweise wurde für Audio-CDs ein α von etwas mehr als 2,2 gewählt, sodass sich eine Abtastfrequenz von 44,1 kHz ergibt. Da sich Sprache größtenteils unter 10 kHz abspielt, wären auch kleinere Abtastraten denkbar. Das Ergebnis der Abtastung ist ein zeitlich aufgelöstes Amplitudendiagramm, wie in 2.2 angedeutet [Stad 09, S. 45-48].

Da Audiodaten im Zeitbereich aufgelöst werden. erhält man pro Auflösungseinheit (in Sekunden, abhängig von der Abtastrate) einen Energie- oder Amplitudenwert. Dieser stellt prinzipiell nur eine relative Lautstärke dar. Die Feinheiten der Sprache spielen sich jedoch im Frequenzbereich ab. Um Audiodaten in den Frequenzbereich zu überführen, wird die Fouriertransformationen verwendet. Da sich Sprachsignale in kurzen Zeitintervallen annähernd periodisch verhalten, erhält man bei der Wahl kleinerer Bereiche für die Transformation einen höher auflösten Frequenzbereich. Wählt man sie zu klein aus, so geht potenzielle Frequenzinformation verloren. Die Bereichsgröße für die Transformation ist diesbezüglich abzuwägen. Des Weiteren ist zu beachten, dass wenn ein Audiosignal in solche Bereiche oder Fenster (Window) zerstückelt wird, dies im Frequenzbereich Störsignale (Leakage) verursacht. Ein solches Fenster kann als Breitbandfilter, also Rechteckfilter, interpretiert werden. Um dem entgegenzuwirken, werden auf Fenster statt Rechteckfilter beispielsweise Hamming-Filter angewendet. Dadurch entstehen Hamming-Fenster, die im Frequenzbereich an den Rändern weniger „verschmieren“ [Pfis 08, S. 45-50].

Zur Verarbeitung der gesamten Länge eines Audio-Signals werden beispielsweise Hamming-Fenster mit fester Fenster-Größe (window size) und mit einer bestimmten Schritt-Größe (window step) über das gesamte Audiosignal gelegt. Man nennt diese Form der Verarbeitung Sliding-Window. Dabei kann die Schritt-Größe geringer sein als die Fenster-Größe, was zur Folge hat, dass mehrere Fenster überlappen und Signal-Information teilen. Wählt man Fenster-Größe und Schritt-Größe geeignet aus, so kann man trotz Überlagerung Mehrinformationen aus den Daten erhalten, da sich das Gesprochene ausreichend stark vom vorhergehenden Fenster unterscheidet. Zur Erzeugung von Frequenzinformationen werden Fourier-Transformationen genutzt. Zur Überführung des Signals in Frequenzen pro Frame wird die **Short-Time Fourier Transformation (STFT)** genutzt. Exemplarisch zeigt Abbildung 2.3 ein Sliding-Window mit einer Fenster-Größe von 25ms und einer Schritt-Länge von 10ms, das heißt, in jedem Fenster außer dem ersten überschneiden sich 15ms Tonmaterial. Darunter zu sehen sind die Frequenzinformationen aus den ersten zwei Fenstern, die mit STFT berechnet wurden.

Ein Spektrogramm, wie in Abbildung 2.5 links, vereint mehrere Fenster in ein Diagramm als Heatmap und wird vereinfacht als Visualisierung des Ergebnisses einer STFT genutzt.

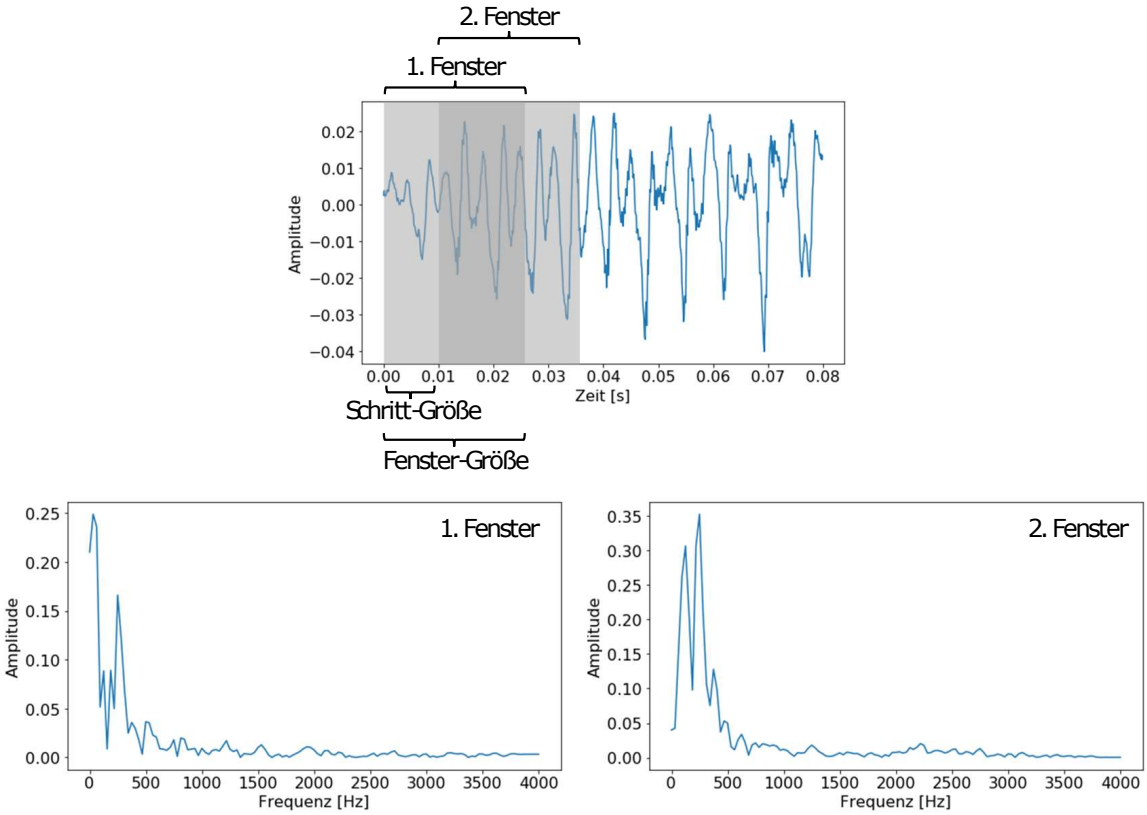

Abbildung 2.3.: Sliding Window mit 25ms Fenster-Größe, 10ms Schritt-Größe, Audiosignal im Zeitbereich (o), Audiosignal von 0 - 25ms (u.l.) und Audiosignal von 10 - 35ms (u.r.) je im Frequenzbereich.

2.2.3. Feature Extraction

Ein grundsätzlicher Schritt einer jeden Mustererkennung ist es, Merkmale (Features) zu erzeugen, die eine Datenmenge möglichst konzis beschreibt. Die Anzahl der Merkmale nennt man Dimensionalität. Die Korrelation einzelner Merkmale sollte möglichst gering sein, um Redundanzen zu vermeiden. Korrelierende Daten können darauf hindeuten, dass es eine gemeinsame zugrunde liegende Ursache gibt, also das eigentliche Merkmal nicht erkannt wurde. Ein weiterer Grund Dimensionalität gering zu halten, liegt in der mathematischen Natur hochdimensionaler Daten. Im Durchschnitt entfernen sich Datenpunkte voneinander, je mehr Merkmale die Daten haben. Das kann es Klassifikatoren oder Clustering-Algorithmen schwierig machen, klare Grenzen oder zusammenhängende Cluster zu erkennen. Diese Problematik wird in der Literatur „Curse of Dimensionality“ genannt [Chan 74]. Ein pragmatischer Grund, die Dimensionalität der Daten möglichst gering zu halten, ist, die Zeit- und Speicherkomplexität mathematischer Modelle realistisch umsetzbar und iterierbar zu halten. Der Prozess der Erzeugung von Features wird Feature Extraction genannt. In der akustischen Sprachverarbeitung haben sich vor allen Dingen folgende zwei Featuresets durchgesetzt.

Mel-Spektogramme basieren auf der Mel-Skala, die nicht-lineare subjektive Wahrnehmung auf eine lineare Skala projiziert, die approximativ mit Formel 2.2 beschrieben wird.

$$f_{mel} = 2959 \cdot \log_{10}(1 + f_{Hz}/700) \tag{2.2}$$

Das Beispiel aus 2.2 wird dadurch annähernd linear lösbar. Die Erhöhung von 100 mel auf 200 mel wird genauso wie die Erhöhung von ca. 1300 mel (1500 Hz) auf 2600 mel (6330 Hz) als Verdopplung der Tonhöhe wahrgenommen. Die Anzahl der Mel-Filter-Bänke wird als Hyperparameter n_{mel} festgelegt. Anschließend wird das Frequenzband zwischen niedrigster f_{min} und höchster f_{max} abzubildender Frequenz auf der Mel-Skala in $n_{mel} + 1$ gleich große Teile zerlegt. Die einzelnen berechneten Teiler auf der Mel-Skala bilden mit ihren jeweils linken und rechten Nachbarn einen Filter. Die Filterform ist frei wählbar, allerdings wird in der Literatur zumeist eine Cosinus- oder Dreiecksform, wie in 2.4 angedeutet, genutzt.

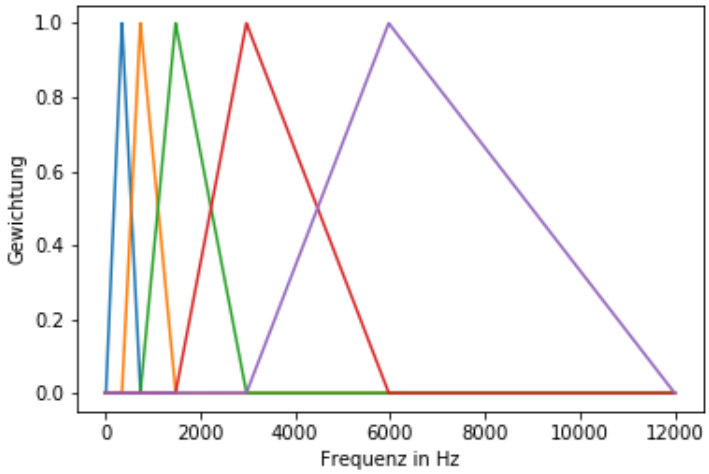

Abbildung 2.4.: Dreieckige Filter in der Mel-Filter-Bank mit $n_{mel} = 5$ und $f_{min} = 20Hz, f_{max} = 12000Hz$.

Jedes Dreieck ist somit ein Bandpass-Filter und wird Mel-Bin genannt. Die Eigenschaft eines jeden Mel-Bins ist, einen bestimmten Hz-Frequenz-Bereich zu gewichten. Aufgrund der Eigenschaft der Mel-Skala werden dadurch niedrigere Frequenzbereiche granularer abgebildet als höhere. Die Hypothese ist, dass in den niedrigeren Frequenzbereichen mehr Informationen über Sprache vorhanden sind, als in den höheren. Ein Nachteil einzelner Mel-Bins der Mel-Filter-Bänke ist, dass sie große Überschneidungen miteinander haben, weshalb eine implizite Korrelation benachbarter Mel-Bins existiert.

Die erzeugte Mel-Filter-Bank wird dann auf jedes Window eines Spektrogramms angewendet. Das Ergebnis ist das Mel-Spektrogramm 2.5 (r). Eine weitere Variante des Mel-Spektrogramm ist das **logarithmierte Mel-Spektrogramm (LogMel)**. Hierbei wird approximativ der Umstand genutzt, dass Lautstärke ebenfalls nicht linear wahrgenommen wird.

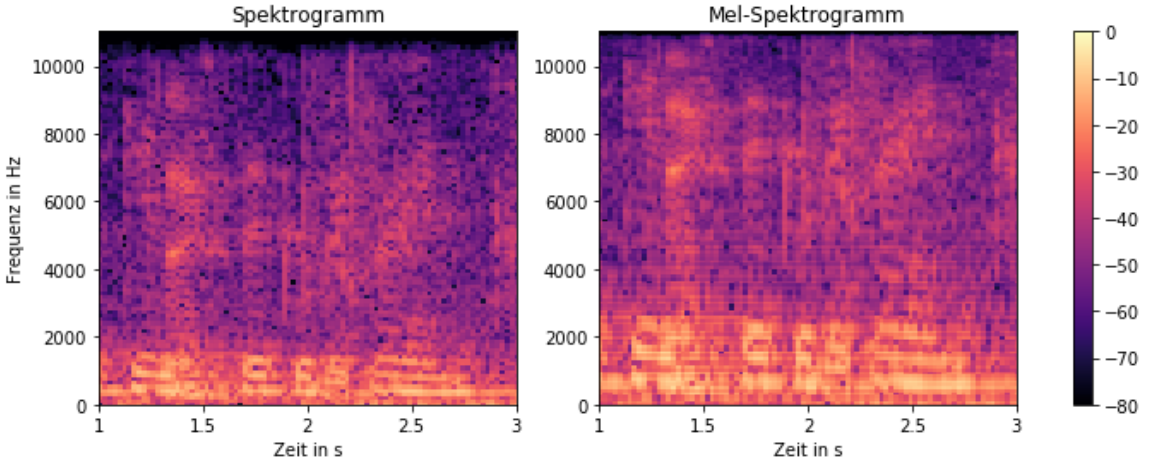

Abbildung 2.5.: Spektrogramm (l), zugehöriges Mel-Spektrogramm (r).

Mel Frequency Cepstral Coefficients (MFCC) sind eine Weiterentwicklung des Mel-Spektrogramms und eine weitere Merkmalsvariante. Um den korrelierenden Effekt der Mel-Filter-Bänke zu vermindern, wird der Logarithmus des Mel-Spektrogramms mit der diskreten Cosinus-Transformation (DCT) verarbeitet. Ein Mel-Cepstrum $c(d)$ kann aus einem Mel-Spektrum $S_m, m = 1, ..., M$ folgendermaßen berechnet werden:

$$c(d) = \frac{1}{M} \sum_{m=1}^{M} (\log(S_m)) \cos\left[d \cdot (m - \frac{1}{2}) \frac{\pi}{M}\right] \Bigg| \quad 0 \leq d \leq D \tag{2.3}$$

D ist frei wählbar und ergibt die Dimensionalität der MFCC. Je mehr Koeffizienten genommen werden, desto größer wird die Abdeckung hoher Frequenzen. Für Sprache ist dies für gewöhnlich nicht nötig und typische Werte für D liegen zwischen 8 und 20. Als Erweiterung der MFCC werden Delta-MFCC und DeltaDelta-MFCC genutzt, die durch Ableitung erster und zweiter Ordnung der ursprünglichen MFCC berechnet werden [Pfis 08, S. 95-97].

2.2.4. Modeling

Für Diarization notwendig sind ausschließlich die Ausschnitte eines Audiosignals, die Sprache enthalten. Der Ablauf zur Erzeugung bereinigter Signale ist Teil der Speaker Segmentation oder Speaker Change Detection. Es zeigt sich, dass Diarization-Systeme wesentlich schlechter abschneiden, wenn sie mit „Nicht-Sprache“, also beispielsweise Sprechpausen, Stille, Störgeräuschen, Musik oder Umgebungsgeräuschen ohne Sprache trainiert werden. Je nach Definition wird zudem zwischen Sprache und Stimme unterschieden, denn Geräusche wie Grunzen oder Schreien werden zwar mit Stimme erzeugt, würden aber nicht zur Sprache gehören. Aus diesem Grund nennt die Literatur zwei Systeme zur Erzeugung „reiner“ Sprachsignale: Speech Activity Dettection (SAD) und Voice Activity Detection (VAD). Im weiteren

Verlauf wird von SAD gesprochen [Sahi 12]. Diese Systeme sollen Sprachaufnahmen vorab segmentieren und sind getrennt von der Aufgabe der Speaker Diarization zu betrachten. So ist es nicht ungewöhnlich, Systeme mit „Oracle“ SAD zu trainieren, indem vorliegende Kenntnisse über echte Sprachsegmente mit genutzt werden.

Der nächste Schritt der Diarization-Pipeline beschäftigt sich mit der Auswahl und Anpassung geeigneter mathematischer Modelle für die Segmentierung und das Clustering der Featuresets und damit den Sprechenden. Als Segmentierung wird ähnlich wie bei SAD die Segmentierung des Featuresets in bestimmte Kategorien verstanden. Während bei SAD zwischen Sprache/Nicht-Sprache segmentiert wird, so wird in diesem Schritt in die einzelnen Sprecheridentitäten segmentiert [Angu 06, S. 13]. Für diese Aufgabe existieren viele verschiedene Methoden. In Prinzip geht es darum, die Ähnlichkeit von Features hervorzuheben oder zu erzeugen. Diese ähnlichen Features gilt es dann durch Clustering-Algorithmen zusammenzuführen.

Eine andere Gruppe von Modellen nutzt Neuronale Netze (NN), die implizit generischer und komplexer sind und größere Datenmengen benötigen. Neuere Methoden mit NN versuchen Speaker Segmentation und Clustering in einem Schritt zu vollziehen.

Vielen Diarization Systemen gemeinsam ist allerdings, dass aus dem Featureset zuerst mithilfe von NN Embeddings erzeugt werden. Es ist nicht ungewöhnlich vortrainierte NNs für die Embedding Extraction zu nutzen. Ebenso gibt es viele Methoden, die unter anderem unterschiedliche Formen von NN nutzen, um Embeddings zu erzeugen. Ein Embedding ist ein Mapping von einer diskreten Variable auf einen Vektor mit kontinuierlichen Werten. NN können so strukturiert und trainiert werden, sodass Embeddings eine implizite kontextabhängige Gruppierung ermöglichen, sich also ähnliche Features im Embedding-Raum nah sind. Zudem werden sie verwendet, um Dimensionalität zu reduzieren [Rouv 15]. Embeddings aus NN im Bereich der Speaker Diarization werden D-Vektoren (d-vectors) oder X-Vektoren (x-vectors) genannt.

Spezifische Modelle werden in den jeweiligen Kapiteln erläutert, insofern sie genutzt worden sind.

Zum Testen der Güte des Diarization Systems werden beim Prozess der Inferenz mit dem System Testdaten verarbeitet, die zuvor nicht genutzt wurden, damit das System sich nicht darauf spezifisch vortrainiert wird und dadurch womöglich das Ergebnis verfälscht wird. Anschließend werden die Ergebnisse mit den Referenzdateien verglichen und bewertet.

2.2.5. Metriken

Zur Bewertung der Qualität und Performance und zum Vergleich verschiedener Modelle sind aussagekräftige Metriken notwendig. Im Bereich der Speaker Diarization werden eine Reihe

spezifischer Evaluations-Metriken verwendet, die in [Fisc 06] festgelegt wurden und wie folgt definiert sind:

$s :=$ ein Segment aus der Gesamtheit zu betrachtender Segmente S
$\mathrm{dur}(s) :=$ die Dauer eines Segments
$N_{ref}(\mathrm{s}) :=$ die Anzahl der Sprecher in der Referenz in s
$N_{sys}(\mathrm{s}) :=$ die Anzahl der Sprecher aus dem Diarization-System (Modell) in s
$N_{correct}(\mathrm{s}) :=$ die Anzahl der Sprecher aus der Referenz in s, für die ein passender gemappter Sprecher in s ebenfalls gesprochen hat
$T_{total} = \sum_{s=1}^{S} dur(s) N_{ref} :=$ die Gesamtdauer der Sprechzeit

Speaker Error (SE) ist die Sprechzeit, die einem falschen Sprecher zugeordnet wurde.

$$E_{spkr} = \frac{\sum_{s=1}^{S} \mathrm{dur}(s)(\min(N_{ref}(s), N_{sys}(s)) - N_{correct}(s))}{T_{total}} \tag{2.4}$$

False Alarm (FA) ist der Anteil an der Gesamtdauer der Sprechzeit, dem fälschlicherweise ein Sprecher zugeordnet wurde, obwohl in der Referenz keine Sprache auftaucht. Die Berechnung wird nur über die Segmente durchgeführt, die in der Referenz als Nicht-Sprache gelabelt wurden.

$$E_{FA} = \frac{\sum_{s=1}^{S} \mathrm{dur}(s)(N_{sys}(s) - N_{ref}(s))}{T_{total}} \tag{2.5}$$

Missed Speech (MS) ist der Anteil an der Gesamtdauer der Sprechzeit, der fälschlicherweise als Nicht-Sprache erkannt wurde, in der Referenz allerdings einen Sprecher aufweist.

$$E_{miss} = \frac{\sum_{s=1}^{S} \mathrm{dur}(s)(N_{ref}(s) - N_{sys}(s))}{T_{total}} \tag{2.6}$$

Overlap speaker (E_{ovl}) ist der Anteil an der Gesamtdauer der Sprechzeit, in der Sprecher in einem Segment keinem Sprecher zugeordnet wurden. Nicht zugeordnete Sprecher aus der Referenz werden in FA und aus dem Diarization-System in MS eingerechnet. Tauchen mehrere Sprecher sowohl in der Referenz als auch im System auf, wird der Fehler in SE zugeordnet [Angu 06].

Diarization Error Rate (DER) ist die Summe der vier Fehlerraten 2.4, 2.5 und 2.6 und wird primär als Evaluations-Metrik herangezogen. DER kann Werte über 1 annehmen.

$$\begin{aligned} DER &= E_{spkr} + E_{FA} + E_{miss} \\ &= \frac{\sum_{s=1}^{S} \mathrm{dur}(s)(\max(N_{ref}(s), N_{sys}(s)) - N_{correct}(s))}{\sum_{s=1}^{S} \mathrm{dur}(s) N_{ref}} \end{aligned} \tag{2.7}$$

Jaccard Error Rate (JER) ist der DER ähnlich. Der Unterschied ist, dass der Anteil eines jeden Sprechers gleich verteilt in die Rate einfließt, unabhängig davon, wie viel Sprache

ein Sprecher tatsächlich produziert hat. JER kann niemals über 1 liegen.
ref := ein spezifischer Sprecher aus der Referenz

$$JER_{ref} = \frac{E_{FA} + E_{miss}}{T_{total}} \tag{2.8}$$

$$JER = \frac{1}{N} \sum_{ref} JER_{ref} \tag{2.9}$$

Des Weiteren werden auch in der ML übliche Metriken wie Precision, Recall, Accuracy und Tabellen wie die Konfusionsmatrix verwendet, um einen Überblick über die allgemeine Performance eines Systems zu geben. Sie werden vor allen Dingen bei binären Klassifikationen verwendet, wie beispielsweise Sprache/Nicht-Sprache (positiv/negativ). In einer Konfusionsmatrix werden Daten der Grundmenge (Actual condition) und die Vorhersagen eines Modells (Predicted condition) eingetragen. Die Tabelle 2.2.5 stellt diese Zusammenhänge dar.

		Predicted condition	
		Positive (PP)	**Negative (PN)**
Actual condition	**Positive (P)**	True Positive (TP)	False Negative (FN)
	Negative (N)	False Positive (FP)	True Negative (TN)

Tabelle 2.1.: Konfusionsmatrix

Precision (prec) ist der Anteil korrekt vorhergesagter positiver Objekte TP zu allen vorhergesagten positiven Objekten $PP = TP + FP$.

$$prec = \frac{TP}{TP + FP} = \frac{TP}{PP} \tag{2.10}$$

Recall (rec) ist der Anteil korrekt vorhergesagter positiver Objekte TP zu allen echten positiven Objekten $P = TP + FN$.

$$rec = \frac{TP}{TP + FN} = \frac{TP}{P} \tag{2.11}$$

Accuracy (acc) ist der Anteil korrekt gefundener positiver und negativer Objekte $TP + TN$ zur Gesamtheit aller Objekte $TP + FN + FP + TN$.

$$acc = \frac{TP + TN}{TP + FN + FP + TN} \tag{2.12}$$

Kapitel 3.

Diarization Systeme

Dieses Kapitel konkretisiert die Ansprüche an ein Diarization System im Kontext dieser Arbeit. Es wird ein grober Überblick über bestehende Diarization Systeme gegeben. Anschließend werden diese hinsichtlich der Ansprüche bewertet und ausgewählt.

3.1. Baseline

Hier wird als Baseline ein Diarization System verstanden, dass als Benchmark und Vergleichsgegenstand dient. Genutzt werden etablierte Methoden, die es mit einem Test- oder Referenzsystem zu schlagen gilt.

Die ausgewählte Baseline ist ein vordefiniertes Rezept für Diarization auf dem AMI-Korpus aus Kaldi. Es befindet sich in `kaldi/egs/ami/s5c`. Dort sind alle notwendigen Skripte vorhanden, um ein vollständiges AMI-Diarization-Experiment durchzuführen. Bei der Umsetzung ist darauf zu achten, den eval2000-Korpus als Validierungsset einzupflegen.

Die erste Stufe der Diarization Pipeline der Baseline wird aus der Abbildung 3.1 dargestellt.

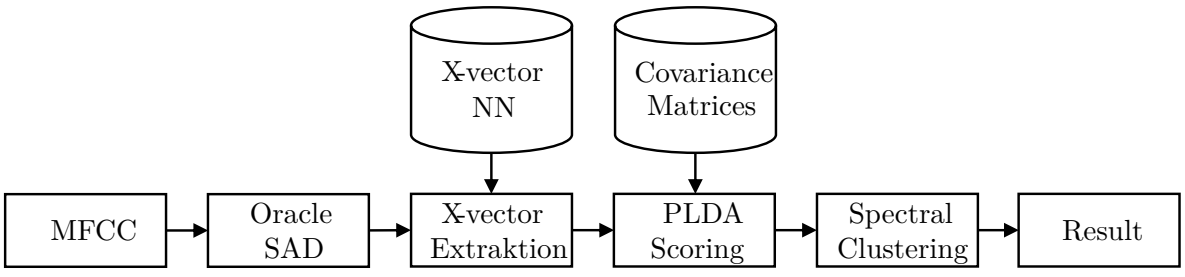

Abbildung 3.1.: Angedeutetes Systemdiagramm der ersten Stufe der Baseline.

Das Rezept ist zweistufig aufgebaut. Zunächst wird Diarization ohne Overlap durchgeführt. Als Featureset werden MFCC genutzt, die mit Oracle SAD in ein vortrainiertes NN gegeben

werden, um X-Vektor Embeddings zu gewinnen. Ein PLDA-Modell wird durch paarweise Ähnlichkeit zwischen allen Embeddings eines Segments trainiert. Das trainierte PLDA-Modell scored anschließend die X-Vektoren des Evaluationssets. Diese Scores werden abschließend mit Spectral Clustering nach [Park 20] geclustert. Die verwendete Form des Spectral Clusterings unterscheidet sich vom Paper, als dass eine normalisierte Laplace-Matrix verwendet wird.

Im zweiten Teil wird ein TDNN-LSTM auf Overlap Detection antrainiert.

3.2. Auswahlverfahren

Die Bewertung von geeigneten Diarization Systemen geschieht unter Berücksichtigung der genannten Ziele in Kapitel 1.2. Zusammengefasst sollen folgende Ansprüche beachtet werden:

1. **Diarization:** Das System segmentiert eine Tonspur nach ihren Sprecheridentitäten.
2. **Overlap:** Das System ist in der Lage, mit Bereichen umzugehen, die Overlap Speech enthalten. Das heißt, es muss mit mehreren gleichzeitig Sprechenden umgehen können.
3. **Komplexität:** Das System ist auf seine Komplexität zu überprüfen. Komplexität wird definiert durch Anzahl verschiedener genutzter Modelle zum Erreichen von Diarization mit Overlap.
4. **Echtzeitfähigkeit:** Das System wird darauf überprüft, ob es Inferenz in Echtzeit, also ohne merkbare Verzögerung, verrichten kann.

Hinzu kommen Qualitätsansprüche, die mehr subjektiver Natur sind und die Bereiche Entwicklung, Dokumentation und Vergleichbarkeit betreffen. Es wird beurteilt, wie leicht sich Code wiederverwenden oder entwickeln lässt und damit Experimente reproduziert werden können und ob ausreichend Dokumentationen vorhanden ist, um Nachvollziehbarkeit zu erleichtern. Zudem wird beachtet, wie gut sich Systeme vergleichen lassen. Nebst hochwertiger Diarization Systeme ist es genauso wichtig, wie leicht diese (aus der Sicht eines Anfängers) nachvollzogen und umgesetzt werden können, damit darauf aufgebaut werden kann.

Das Forschungsgebiet der ASR und Diarization ist weitläufig und es gibt einige regelmäßig stattfindende Konferenzen und Challenges. So bietet sich beispielsweise die DIHARD Speech Diarization Challenges als Quelle sehr gut an, da das Evaluationset bei allen Teilnehmern das gleiche ist, also die Vergleichbarkeit gegeben ist. Zum Zeitpunkt der Evaluation von Systemen war die aktuellste Challenge die zweite DIHARD Challenge. Die Ziele waren automatische Detektion und Labeling aller Segmente von Sprechenden. Das heißt, es sollen die Anzahl der Sprechenden und für jeden Sprechenden alle korrespondierenden Segmente herausgefunden

werden [Ryan 21]. Die Tabelle 3.1 zeigt die ersten drei Platzierten auf Track 2 (Diarization ohne Oracle SAD)[1], die genauer analysiert werden.

		Core set		Full set	
Platz	**Team**	**DER**	**JER**	**DER**	**JER**
1	USTC-NELSLIP	19.37	39.22	16.78	34.42
2	dh_Hitachi-JHU	20.01	41.78	16.94	36.31
3	DKU-Duke-Lenovo	21.63	41.80	18.90	36.69

Tabelle 3.1.: 2. DIHARD Challenge Track 2: Top 3 Ergebnisse.

3.2.1. USTC-NELSLIP DIHARD III System

Das Team um **USTC-NELSLIP** (University of Science and Technology of China - National Engineering Laboratory for Speech and Language Information Processing) erstellte ein Diarization System aus sechs Modulen, die in [Wang 21b] beschrieben wurden.

1. **Audio Domain Classification:** Ein ResNet wurde antrainiert, um Audio in 11 Domänen einzuordnen. Das Netzwerk erzielte 100 % Accuracy und wurde auf das Evaluationsset angewendet.

2. **Speech Enhancement**: Es wurden mehrere Methoden verglichen und verwendet, um Sprache in bestimmten Bereichen zu verstärken. Unter anderem wurden Neuronale Netze mit LSTM verwendet.

3. **Clustering Based Diarization System:** Die Pipeline war folgende: SAD, X-Vektor Extraktion, Probabilistic Linear Discriminant Analysis, Agglomerative Hierarchical Clustering, Variational Bayes-Hidden Markov Model. Für SAD wurden drei unterschiedliche NN antrainiert.

4. **Iterative Speech Seperation basierende Diarization Systeme:** Es wurde ein vortrainiertes NN für die Seperation und ein NN für SAD genutzt. Alle SAD Ergebnisse kombiniert zeigten die Overlap Regionen an. Iterativ wurde es dadurch, dass das Seperation-NN zweimal adaptiv neu trainiert wurde. Zuerst wurden die Ergebnisse des Schritt 3 verwendet. Anschließend wurde dieses Modell anstelle des vortrainierten NN verwendet, um noch einmal Diarization zu machen. Diese Ergebnisse wurden genutzt, um das vortrainierte NN endgültig abzustimmen.

5. **Iterative Target-Speaker (TS) - VAD basierende Diarization Systeme:** TS-VAD nutzt LogMel Input zusammen mit I-Vektoren und macht Vorhersagen über Sprachaktivität pro Frame für alle Sprecher gleichzeitig. Auch hier wurde eine iterative Methode genutzt, um die Feinabstimmung des TS-VAD Modell vorzunehmen.

[1] https://dihardchallenge.github.io/dihard3/results#datatable_visualization1

6. **Post-processing:** Die Domain Klassifikation wurde genutzt, um die am besten abgestimmten Modelle für die entsprechenden Domänen zu verwenden. Des Weiteren wurde Diarization Output Voting Error Reduction (DOVER) - Lap verwendet, um die Ergebnisse der vielen verschiedenen Diarization Systeme zu fusionieren.

Zwischenfazit: Es wurden eine große Anzahl verschiedener Methoden, Modelle und Algorithmen verwendet. Es wurde kein Repository angegeben. Vereinzelt sind Repositorys in verlinkter Literatur zu finden (zum Beispiel DOVER-Lap). Es wurden Angaben über Inferenzzeiten gemacht.

3.2.2. Hitachi-JHU DIHARD III System

Das Team um **Hitachi** und der **John Hopkins University (JHU)** verwendete fünf Diarization Systeme mit DOVER-Lap, die in [Hori 21] beschrieben wurden.

1. **VAD:** Zwei VAD-Systeme wurden verwendet. Eins basierte auf SincNet VAD aus dem pyannote Framework, das andere auf Time-Delay Neural Network (TDNN) und stammt aus deinem Kaldi Rezept.

2. **2 X-Vektor basierende Systeme:** Eins basiert auf dem TDNN X-Vektor basierten Extractor des Gewinners von DIHARD II mit anschließendem VB-HMM x-vectors (VBx) Clustering. Das zweite auf vier unterschiedliche Res2Net basierenden X-Vektor Extractor des Gewinnersystems der VoxCeleb Speaker Recognition Challenge 2020, deren Ergebnisse mit einem angepassten DOVER-Lap kombiniert wurden.

3. **2 EEND basierende Systeme:** Beide sind Erweiterungen des ursprünglichen SA-EEND mit variabler Anzahl an Sprechenden und verbesserter Inferenz. Eins ist EEND-EDA, das andere SC-EEND.

4. **Hybrides Subsystem:** Die Diarization Ergebnisse des TDNN X-Vektor mit VBx Clustering wurden mit EEND nach verarbeitet.

5. **Systeme zusammenführen:** Mit DOVER-Lap wurden alle Diarizationsysteme zusammengeführt.

Zwischenfazit: Es wurden eine überschaubare Anzahl verschiedener Methoden, Modelle und Algorithmen verwendet. Besonders interessant ist EEND, da mit einem Modell fast ein vollständiges Diarization-Modell zu bewerkstelligen ist. Es ist ein Repository für EEND-EDA angegeben. Vereinzelt sind Repositorys in verlinkter Literatur zu finden. Es wurden öffentlich zugängliche Softwarekomponenten und Skripte wiederverwendet. Es wurden Angaben über Inferenzzeiten gemacht.

3.2.3. DKU-Duke-Lenovo DIHARD III System

Das Team um **Duke Kunshan University (DKU) - Duke-Lenovo** teilte das Datenset in zwei Gruppen auf und konzipierte spezialisierte Systeme für jedes Set. Das System wurde in [Wang 21a] beschrieben.

1. **VAD:** Das VAD-System basiert auf einem ResNet mit Global Averaging Pooling, einem 2-stufigen Bi-LSTM.
2. **Speaker Embedding Extraction:** Genutzt wird ein 3-stufiges System basierend auf ResNet, bestehend aus Frontend Pattern Extractor, einem Encoder Layer und einem Backend Klassifikator.
3. **Segmentation:** Die Embeddings werden uniform segmentiert, anschließend rekursiv paarweise zusammengeführt, solange die Cosinus-Ähnlichkeit geringer als 0.6 war.
4. **Similarity Measurement and Clustering:** Auf einem Teil der Daten wurden Attention-basierende NN genutzt, um die Ähnlichkeit zweier Segmente zu berechnen. Auf den anderen Teil wurden Kosinus-Ähnlichkeit mit anschließender Agglomerative Hierarchical Clustering (AHC) - Segmentierung verwendet.
5. **TSVAD:** Ein TSVAD Modell wird verwendet, um die Ergebnisse zu verfeinern.

Zwischenfazit: Die Anzahl der Methoden, Modelle und Algorithmen sind gering. Es wurde kein Repository angegeben. Vereinzelt sind Repositorys in verlinkter Literatur zu finden. Es wurden keine Angaben über Inferenzzeiten gemacht.

3.3. Ergebnis

USTC-NELSLIP und dh_Hitachi-JHU nutzen viele verschiedene State of the Art Methoden, um sie anschließend mit DOVER-Lap zusammenzuführen und in Prinzip das Beste aus allen Modellen herauszuholen. Dem inhärent ist allerdings eine wesentlich größere Komplexität. Ein Sub-System, nämlich EEND-EDA, wird zusätzlich betrachtet, da es alleine bereits Diarization mit Overlap ermöglicht. Die Echtzeitfähigkeit wird naiv derart bewertet, ob das Verhältnis $\frac{t_{inferenz}}{t_{dur}} \leq 1$ ist. Ein weiteres Kriterium müsste sein, wie groß ein Verarbeitungssegment (zum Beispiel 1 s) im Verhältnis zur Verarbeitungszeit sein kann. Je geringer beide Werte bei guter DER sein können, desto besser. Allerdings sind diese Informationen nicht gegeben.

Sekundär werden das Vorhandensein von Repositorys oder Verweise auf genutzte Frameworks und Rezepte sowie Dokumentation und technische Erklärungen positiv bewertet. Die Bewertungsmetrik geht von `++` (sehr positiv, +2) über `0` bis `-` (sehr negativ, -2). Das Fehlen einer Bewertungsgrundlage (zum Beispiel Fehlen von Inferenzzeiten) wird mit „NA“ bewertet, was

implizit gleich bedeutend ist mit „sehr negativ“. Die Bewertung geschieht in Relation zum jeweils best Abschneidenden der jeweiligen Kategorie, ist allerdings überwiegend qualitativer Natur. Das Endergebnis ist die Summe der Bewertungen mit gleicher Gewichtung. Die Ergebnisse sind der Tabelle 3.2 zu entnehmen.

	Primär					Sekundär			
System	DER	Overlap	Komplexität	Online	Σ_1	Repo	Doku	Σ_2	Σ
USTC-NELSLIP	++	++	−−	−−	**0**	−−	−−	**-4**	**-4**
dh_Hitachi-JHU	++	++	−	−−	**1**	0	0	**0**	**1**
DKU-Duke-Lenovo	+	++	0	NA	**1**	−−	−−	**-4**	**-3**
EEND-EDA	0	++	+	−−	**1**	++	+	**3**	**4**

Tabelle 3.2.: Bewertung der Diarization Systeme hinsichtlich der Ziele und Umsetzbarkeit.

Gerade im Hinblick auf die Realisierbarkeit im Umfang dieser Arbeit stellen Systeme der DIHARD Challenge ein erhebliches Problem dar, da die besten Systeme sehr umfangreich und komplex aufgebaut sind. Auch wenn EEND-EDA mit den DOVER-Lap kombinierten Systemen nicht mithalten kann, so ist der Ansatz eines einzelnen Modells zur Erzeugung von Overlap-fähiger Diarization mit variabler Anzahl Sprechender ein reizvoller Grund, dieses umzusetzen und zu testen. Darüber hinaus werden dazu nutzbare Repository und Dokumentation angeboten, was den Einstieg vereinfachen sollte. Aus diesen Gründen wird als Testsystem EEND-EDA umgesetzt.

Kapitel 4.

End-to-end neural network-based speaker diarization model (EEND) mit Encoder-Decoder-Attractor (EDA)

Als Grundlage dieser Abschlussarbeit wird die Diarization-Methode Self-Attentive (SA)-EEND mit EDA von [Hori 20] verwendet. Im Folgenden werden ihre Herangehensweisen und Methodiken erklärt. Diese Erweiterung baut unmittelbar auf die Systeme von [Fuji 19a] und [Fuji 19b] auf. EEND-EDA unterscheidet sich von typischen Diarization Systemen folgendermaßen:

- Kann implizit mit Overlap umgehen.
- Vereinfachung durch Wegfallen expliziter Module für:
  - SAD
  - Speaker Identifcation
  - Clustering
- Implizite Verminderung der DER über eine Loss-Funktion.
- Kann mit variabler Anzahl von Sprechenden umgehen.

Die Erweiterung durch Horiguchi et al. ist in erster Linie EDA. EEND muss mit einer fixen Anzahl von Sprechenden während der Inferenz initialisiert werden. EDA weicht diese Bedingung auf und ermöglicht eine dynamische Berechnung der Anzahl von Sprechenden, die für die Inferenz genutzt werden kann.

4.1. Datensatz des Papers

In ihrer experimentellen Umsetzung wurden Datensets aus simulierte Mixe aus Switchboard-2 (Phase I, II, III), Switchboard Cellular (Teil 1 und 2), der NIST Speaker Recognition Evaluation (2004, 2005, 2006, 2008) und der MUSAN Korpus für Lärm und Geräusche mit simulierten Nachhall verwendet. Damit stammte der Großteil der Trainings- und Evaluationsdaten aus der Domäne der Telefonie. Aus den genannten Korpora wurde ein klar definierter Korpus

erzeugt mit 1, 2, 3 und 4 Sprechenden mit ähnlichen Overlap-Anteilen von circa 34 % wie beispielsweise die 2-Sprecher Mixturen. Zur Evaluation kommen noch CALLHOME, der Corpus of Spontaneous Japanese und das zweite DIHARD Datenset hinzu.

4.2. Funktionsweise SA-EEND

Als Input erhält das SA-EEND T-lange LogMel-Features, welche durch Transformer-Encoder je Zeitslot t auf ein Embedding $e_t \in \mathbb{R}^D$ projiziert werden. Diese werden elementweise mit einer Sigmoid-Funktion $f : \mathbb{R}^D \to \mathbb{R}^S$ verrechnet, um Posterior $\hat{y}_t = [\hat{y}_{t,1}, ..., \hat{y}_{t,S}]^T \in (0,1)^S$ für Sprechende zum Zeitslot t zu erhalten. In der Trainingsphase wird EEND mit Permutation Invariant Training (PIT)[Yu 17] optimiert, das heißt, Loss wird zwischen Posterior und Groundtruth Label wie folgt berechnet:

$$L_d = \frac{1}{TS} \underset{\phi \in perm(1,...,S)}{\operatorname{argmin}} \sum_{t=1}^{T} H(y_t^{\phi}, \hat{y}_t), \tag{4.1}$$

wo perm(1, ..., S) für das Set für alle Permutationen von Sprechern steht. $H(y_t^{\phi}, \hat{y}_t)$ ist die Binary Cross Entropy und festgelegt als:

$$H(y_t, \hat{y}_t) := \sum_s -y_{t,s} \log \hat{y}_{t,s} - (1 - y_{t,s}) \log (1 - \hat{y}_{t,s}). \tag{4.2}$$

4.3. Funktionsweise EDA

Da SA-EEND bei der Inferenz durch die lineare Transformation begrenzt ist durch die Anzahl Sprechenden S, enwickelte Horiguchi et al. EDA um Attractors aus der Embedding-Sequenz zu bestimmen. Die oben genannten D-dimensionalen Embeddings werden in einen unidirektionalen LSTM-Encoder gepseist, durch den der finale Hidden State Embeddings $h_0 \in \mathbb{R}^D$ und der cell state $c_0 \in \mathbb{R}^D$ berechnet wird:

$$h_0, c_0 = \text{Encoder}(e_1, ..., e_T). \tag{4.3}$$

Anschließend wird ein Zeit-invarianter D-dimensionaler Attractor $(a_s)_s$ mit einem unidirektionalen LSTM-Decoder mit den Initialwerten h_0 und c_0 berechnet:

$$h_s, c_s, a_s = \text{Decoder}(h_{s-1}, c_{s-1}, 0) \tag{4.4}$$

Theoretisch lassen sich dadurch unendlich viele Attractors berechnen. Die Abbruchfunktion ist in einem Fully-connected Layer mit einer Sigmoid-Funktion gemäß

$$p_s = \frac{1}{1 + \exp(-(w^T a_s + b))} \tag{4.5}$$

realisiert, wo w weights und b bias trainierbar sind.

Die Groundtruth Labels $l = [L_1..., l_{S+1}]^T$ der Trainingsphase mit Anzahl Sprechende S ist definiert als:

$$l_s \begin{cases} 1, & (s \in \{1, ..., S\}) \\ 0, & (s = S + 1). \end{cases} \tag{4.6}$$

Der Attractor Existence Loss L_a zwischen Labels und Schätzung $p = [p_1, ..., p_{S+1}^T$ wurde mit der Binary Cross Entropy aus 4.2 wie folgt berechnet:

$$L_a = \frac{1}{1 + S} H(l, p). \tag{4.7}$$

War während der Inferenz die Anzahl Sprechenden S gegeben, so werden die ersten S Attractors genutzt. War S unbekannt, wird mit einem Grenzwert τ die Anzahl $\hat{S}$ Attractors geschätzt:

$$\hat{S} = \max\{s \mid s \in \mathbb{Z}_+ \wedge p_s \geq \tau\} \tag{4.8}$$

[Hori 20].

Die Abbildung 4.1 gibt einen Überblick über das SA-EEND EDA Diarization System und die besprochenen Komponenten.

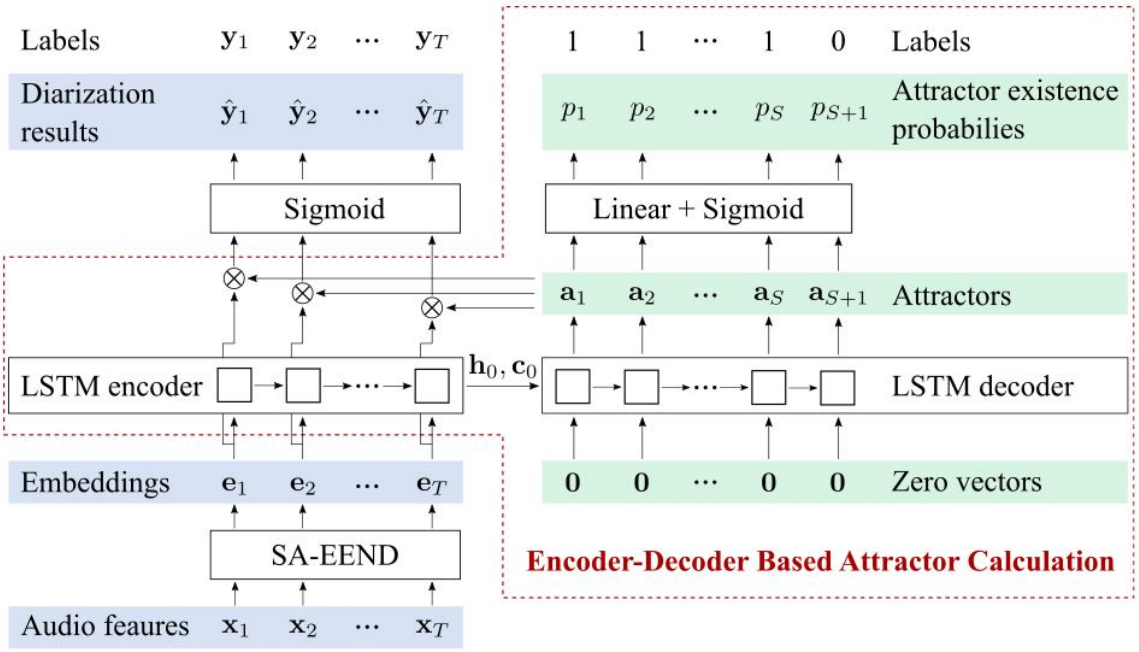

Abbildung 4.1.: SA-EEND mit EDA Überblick [Hori 20, S. 2, Fig. 1].

Kapitel 5.

Umsetzung

Dieses Kapitel erklärt den experimentellen Aufbau und Umsetzung der Baseline und SA-EEND mit EDA. Dafür wird auf die verwendete Hardware und Software und die genutzten Datensätze eingegangen. Anschließend werden die Ergebnisse der durchgeführten Experimente vorgestellt. Die Umsetzung erfolgte auf einen Fork vom Repository EEND-EDA. Näheres ist im Anhang A beschrieben.

5.1. Hardware

Für die Berechnung von Features und der Modelle wird ein Hochschulserver verwendet, dessen technische Spezifikation folgende ist:

Prozessoren	2x Intel Xeon Gold 6252, 24 (48) x 2,10 GHz
RAM	377 GB
Grafikkarten	8x GeForce RTX 2080 Ti

Tabelle 5.1.: Auflistung genutzter Hardware.

Da sich diese Arbeit auf Verarbeitungszeiten bezieht, können diese Randdaten eine hilfreiche Ergänzung zur relativen Einschätzung nötiger Ressourcen und ihre Zeiteffizienz sein. Da es sich um einen einzelnen Rechner handelt, steht ein Vergleich mit einem Cluster in keinem Verhältnis, weshalb die angegebenen Zeiten eher im Kontext von hochleistungsfähigen PCs oder einzelnen Servern zu sehen sind. Hinzukommt die Randbedingung, dass der Server von vielen Parteien genutzt wird, insofern niemals die gesamte Server-Kapazität für ein Projekt genutzt wird (werden sollte). Zur Umsetzung dieses Projekts wurde abhängig von akuter Auslastung manuell der Gebrauch von CPU und GPU auf 12 – 30 Threads respektive 1 - 2 Grafikkarten begrenzt. Daraus ergibt sich, dass circa 12,5 % - 25 % der möglichen Ressourcen verwendet wurden. Wo möglich wurde auf Vergleichbarkeit geachtet, jedoch sollte davon abgesehen werden, Zeitwerte aus dieser Arbeit weiterzuverwenden.

5.2. Technische Grundlagen

Dieses Kapitel beschreibt verwendete Software, Frameworks und Dateiformate, die bei der experimentellen Umsetzung genutzt wurden.

5.2.1. Dateiformate

Audioformate und -verarbeitungsprogramme

Das primär genutzt Audioformat ist das WAVE-Dateiformat (.wav). WAV-Dateien enthalten unkomprimierte PCM-Rohdaten, deren Qualität nur von der Abtastrate und Auflösung abhängt. Dieses Format wird bevorzugt genutzt, da jegliche grundlegende negative Beeinflussung bei der Digitalisierung, wie zum Beispiel Qualitätsverlust durch Kompression, zunächst ausgeschlossen werden kann. Dadurch reduziert man potenzielle kaskadierende Fehler bei der Verwendung und Verarbeitung.

Ein hier genutztes Korpus vom Linguistic Data Consortium (LDC) verwendet das National Institute of Standards and Technology Speech Header Resources Format oder kurz NIST SPHERE (.sph). Die ersten 1024 Bytes bestehen aus ASCII key-value Paare mit META-Information, gefolgt von binären Audiodaten. [Jane 91] Das LDC stellt Konvertierungstools[1] zur Verfügung. Genutzt wird `sph2pipe_v2.5`, das eine Reihe von Konvertierungsoptionen anbietet, hier aber nur genutzt wird, um SPHERE Dateien in WAV-Dateien umzuwandeln. Ein weiteres Programm zur Konvertierung und Veränderungen von Audiodateien ist `sox`[2]. Es wird in dieser Arbeit verwendet, um aus Stereo Mono zu mixen und Sampleraten anzupassen.

Diarization Dateien

NIST versuchte Standardisierungen bei der Herangehensweise in der Entwicklung und Evaluierung von ASR-Systemen. Für die Diarization wurde das Dateiformat Rich Transcription Time Marked (RTTM) eingeführt. Es enthält unter anderem Informationen darüber, welche Sprecher wann und wie lang gesprochen haben. [NIST 09] Typischerweise muss die RTTM-Datei aus Transkripte und weiteren Dateien erzeugt werden. Es stellt das zentrale Format für den Vergleich von Diarization-Systemen dar. Bei der Inferenz wird eine RTTM-Datei erzeugt, die dann per Evaluationsskripts mit der Referenz-RTTM-Datei verglichen wird. Tabelle 5.2 zeigt beispielhaft die ersten drei Zeilen einer RTTM-Datei.

[1] https://www.ldc.upenn.edu/language-resources/tools/sphere-conversion-tools
[2] http://sox.sourceforge.net/sox.html

Type	File	Channel	Beginning time (s)	Duration (s)	Ortho	stype	Speaker Name	Conf	SLAT
SPEAKER	en_4156-A	1	301.85	0.63	<NA>	<NA>	en_4156-A	<NA>	<NA>
SPEAKER	en_4156-A	1	304.71	2.01	<NA>	<NA>	en_4156-A	<NA>	<NA>
SPEAKER	en_4156-A	1	307.63	3.53	<NA>	<NA>	en_4156-A	<NA>	<NA>

Tabelle 5.2.: Die ersten drei Zeilen der RTTM-Datei für eval2000.

5.2.2. Kaldi

Das Github-Archiv für EEND-EDA wurde als Rezept für Kaldi konzipiert. Es hat harte Abhängigkeiten zu Kaldi-spezifischen Skripte und Bibliotheken.

Kaldi ist ein kostenfreies Open-Source Toolkit für Speech Recognition und ist hauptsächlich in C++ geschrieben und mit Apache License v2.0 lizenziert. Es stellt Speech Recognition Systeme zur Verfügung, die auf finite-state transducers basieren. Zudem unterstützt es jegliche Form und Kontextgröße akustischer Modelle mit beispielsweise (subspace) Gaussian mixture models ((S)GMMs) und linearen und affinen Transformatoren. Das Ziel des Projekts ist es, flexibel zu sein, indem es einfach modifizierbar und erweiterbar aufgebaut ist. Dies wird unter anderem durch eine Vielzahl von Beispiel-Projekten auf Grundlage unterschiedlicher Korpora erreicht. Diese ermöglichen einen vereinfachten Einstieg und gewähren Einblicke in die Herangehensweise zur Kreierung von ASR-Systemen. Diese Beispiele können dann auf eigene Anforderung modifiziert, erweitert oder zusammengeführt werden. [Pove 11]

Aufbau eines Kaldi-Rezepts

Als ein Rezept in Kaldi wird ein Ordner verstanden, der alle notwendigen Strukturen und Skripte enthält, um ein Experiment im Bereich der ASR durchzuführen. Diese Skripte sind zumeist in `bash`, `perl` und `python` geschrieben und verwenden kompilierte C++ Bibliotheken, aber auch Skripte anderer Rezepte, die als Goldstandard genutzt werden. Abbildung 5.1 visualisiert die Ordnerstruktur und den Inhalt eines typischen Kaldi-Rezepts.

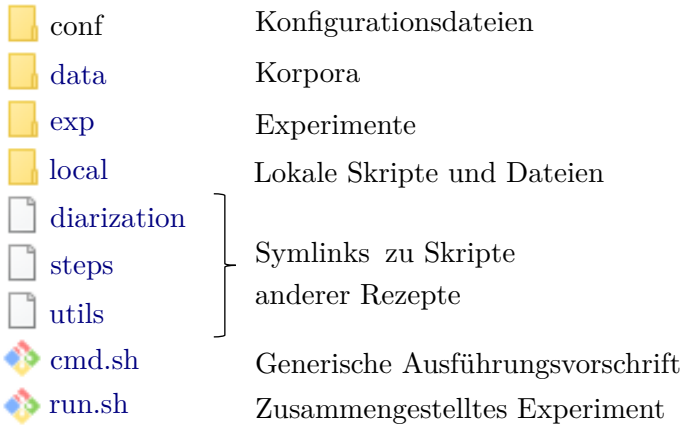

Abbildung 5.1.: Ordnerstruktur eines typischen Kaldi-Rezepts.

Aufbau eines Korpus Häufig auch "Kaldi-Daten"genannt, besteht ein Kaldi-Korpus aus fest definierten Dateien, die für viele Skripte Grundvoraussetzung sind. Kaldi abstrahiert mit dieser Datenstruktur spezifische Korpora. `wav.scp` enthält beispielsweise Verweise auf oder Verarbeitungsvorschriften für Audiodateien des jeweiligen Korpus und muss erzeugt werden. Alle anderen Dateien werden durch Transkripte und Segmentdateien des jeweiligen Korpus hergeleitet. Zu erwähnen ist noch, dass für Diarization ein teils anderes Set an Dateien genutzt wird, als für beispielsweise ASR. Zumeist wird ein kleineres Subset mit weniger Metadaten genutzt, das im Folgenden beschrieben wird.

Die Datei `wav.scp` enthält, wie bereits erwähnt, direkte oder indirekte Verlinkungen zu den Audiofiles.

`wav.scp`

Recording-ID	Filename/Lambda
en_4156-A	`sph2pipe -f wav -p -c 1`

Die Datei `segments` enthält alle Sprachsegmente und ist verlinkt auf die Recording-ID.

`segments`

Utterance-ID	Recording-ID	Segment-Begin	Segment-End
en_4156-A_030185-030248	en_4156-A	301.85	302.48

Die Datei `utt2spk` enthält alle Sprachsegmente mit Verlinkung auf den Sprechenden.

`utt2spk`

Utterance-ID	Speaker-ID
en_4156-A_030185-030248	en_4156-A

Manchmal sind noch `spk2utt` und `reco2dur` nötig. Sie lassen sich mit `utils/utt2spk_to_spk2utt.pl utt2spk > spk2utt` respektive `utils/data/get_reco2dur.sh *` (letzteres Skript nutzt entweder wav.scp oder utt2dur).

5.2.3. GitHub: EEND

Für die Umsetzung dieses Projekts wird primär vom öffentlich zugänglichen Git-Archiv von [Hori 20] über EEND mit EDA ausgegangen. Als Framework-technische Voraussetzung wird Kaldi und als Programmiersprache Python und bash genutzt.

Yusuke Fujita und Shota Horiguchi ermöglichen mit der Veröffentlichung des Programmcodes[3] zu ihren Papern eine wissenschaftliche Verifizierung, Nachvollziehbarkeit und Reproduzierbarkeit ihrer Ergebnisse. Auch nicht zu unterschätzen ist der Lerneffekt für Anfänger in diesem Feld, die sich an professionellen Beispielen Herangehensweisen und Umsetzungen erarbeiten können, die über einführende Tutorials hinaus gehen. Dieses Archiv ist über die Jahre an weiterführenden Papern gewachsen, weswegen im Projekt EEND auch SA und EDA umgesetzt aufzufinden sind.

5.3. Datensatz

Als Trainingsdatensatz wurde Augmented Multi-Party Interaction, als Evaluationsdatensatz eval2000 vorgegeben. Der Trainingsdatensatz AMI unterscheidet sich signifikant von denen in den Arbeiten um Horiguchi et al. Aufgrund der Situation um die Unzugänglichkeit zu den Zieldaten, wurden Ersatzdaten gesucht, die den Kontext von Arzt-Patienten-Gespräche in einem Raum mit relativ nahen Mikrofonen am besten widerspiegeln können. AMI enthält solche Aufnahmen. Die Ergebnisse in EEND-EDA wurde zum größten Teil Telefonaufnahmen und simulierten Aufnahmen erzielt. Das eval2000 Evaluationsset ist als Vergleichskorpus konzipiert und in der Vergangenheit aktiv genutzt worden. Damit kann das Diarization-System mit Systemen verglichen werden, die ebenfalls dieses Evaluationsset genutzt hatten.

5.3.1. Augmented Multi-Party Interaction (AMI)

Die Ziele des AMI-Projekts sind vielfältig. Das AMI-Konsortium verfolgt mit diesem Projekt unter anderem das Ziel, Technologien zu entwickeln, um menschliche Interaktion bei Meetings zu unterstützen und jene Interaktionen strukturiert zu dokumentieren. So werden beispielsweise Präsentationsfolien, Textinformationen, wie handschriftliche Notizen oder Tafelanschriften, aber auch Audiomitschnitte aus unterschiedlichen Quellen im Raum aufgezeichnet. Diese Daten werden zum Beispiel durch Transkripte oder Indizierung und zeitliche Synchronisation der Daten weiterverarbeitet, sodass ein Meeting elektronisch verarbeitbar und der Zugang zu Informationen aus dem Treffen vereinfacht wird. Das AMI-Korpus, welches durch dieses Projekt entstand, besteht aus fast 100 Stunden Meetings mit entsprechenden Video- und Audio-Aufnahmen und viele verschiedene Annotationen und Transkriptionen. Jedes Meeting

[3] https://github.com/hitachi-speech/EEND

wurde mit vier Personen in englischer Sprache durchgeführt. Die meisten Sprecher waren keine Muttersprachler. Audiodaten wurden auf zwei Arten gesammelt. Omnidirektionale Ansteckmikrofone und Headsetmikrofone nehmen Gespräche in direkter Nähe auf und die Gesamtkulisse wurde durch zwei Arrays mit vier und acht omnidirektionalen Mikrofonen aufgefangen. [Mcco 05]

Die Audiodaten lassen sich in unterschiedlicher/n Qualität und Ursprungs herunterladen. Es gibt Szenario (S) und Nicht-Szenario (N) Meetings und vier Aufnahmeräume: IDIAP (I, 1000), Edinburgh (E, 2000), TNO (T, 3000), ISSCO-IDIAP (I, 4000). Jede Szenario-Aufnahmesession wird in vier (a, b, c, d) einstündige Submeetings aufgeteilt. Alle Meetings sind in Tabelle 5.3 aufgelistet. Zu allen Szenario-Meetings gehören separate (a, b, c, d)-Submeetings-Aufnahmen, die in der Tabelle 5.3 nicht dargestellt sind. Die Benennung erfolgt in der Abfolge: Aufnahmeraum (Buchstabe), Szenariotyp, Aufnahmeraum (Zahl) summiert mit Aufnahmesession-Nummer, Aufnahmesession-Untersession.

Beispiel: EN2002b – Edinburgh (E 2000), Nicht-Szenario-Meeting (N), zweiter Aufnahmetag (2), Untersession am Tag 2 (b).

Vorliegend sind Aufnahmen der Meetings im WAV-Format in Mono, 16000 Hz Samplerate und 32 Bit. Jedes Meetings enthält Aufnahmen aus den Arrays, Headsets und Lapels (Ansteckmikrofone). Zudem gibt es je ein Mix aller Headset- und Lapel-Aufnahmen eines Submeetings/Meetings. [Rena 07]

Szenario Meetings			**Nicht-Szenario Meetings**		
Edinburgh	**IDIAP**	**TNO**	**Edinburgh**	**ISSCO-IDIAP**	**IDIAP**
ES2002	IS1000	TS3003	EN2001a	IB4001	IN1001
ES2003	IS1001	TS3004	EN2001b	IB4002	IN1002
ES2004	IS1002	TS3005	EN2001d	IB4003	IN1005
ES2005	IS1003	TS3006	EN2001e	IB4004	IN1007
ES2006	IS1004	TS3007	EN2002a	IB4005	IN1008
ES2007	IS1005	TS3008	EN2002b	IB4010	IN1009
ES2008	IS1006	TS3009	EN2002c	IB4011	IN1012
ES2009	IS1007	TS3010	EN2002d		IN1013
ES2010	IS1008	TS3011	EN2003a		IN1014
ES2011	IS1009	TS3012	EN2004a		IN1015
ES2012			EN2005a		IN1016
ES2013			EN2006a		
ES2014			EN2006b		
ES2015			EN2009b		
ES2016			EN2009c		
			EN2009d		

Tabelle 5.3.: Alle AMI-Sessions aufgelistet.

Die Verteilung der Nicht-Sprache und Sprache kann der Tabelle 5.4 entnommen werden.

			gleichzeitig Sprechende			
Total	**Nicht-Sprache**	**Sprache**	1	2	3	4
100	16.5	83.5	72.3	10.1	1.0	0.05

Tabelle 5.4.: Verteilung von Nicht-Sprache und gleichzeitig sprechenden im AMI Trainingsset in %.

Für die Erstellung der RTTM-Datei und Kaldi-Daten können fertige Skripte verwendet werden. Sie nutzen das Repository https://github.com/BUTSpeechFIT/AMI-diarization-setup, in dem Diarization Referenzdateien, aber auch eine standardisierte Aufteilung des AMI-Datensatzes in Trainings-, Development- und Validierungssets (train, dev, test) vorhanden ist. Das Skript zur Aufbereitung befindet sich in Kaldi bei `egs/ami/s5c/local/prepare_data.py`. Die einzelnen RTTM-Dateien können aus dem erwähnten AMI-Repository wie folgt herausgezogen und zusammengefügt werden:

```
1 for set in train dev test; do
2     cat AMI-diarization-setup/only_words/rttms/${set}/*.rttm \
3         > data/${set}/rttm
4 done
```

Listing 5.1: Formatierung der ursprünglichen RTTM angleichen.

Für die Trainingsphasen der hier genutzten Diarization Systeme werden nicht alle Audiodateien des AMI-Korpus verwendet. Aus jedem Meeting wird jeweils der angebotene Headset-Mix verwendet. Dieser ist, wie der Name andeutet, ein Mix der Aufnahmen der vier Headsets in einem Meeting.

5.3.2. 2000 HUB5 English Evaluation Speech (eval2000)

Das Korpus *2000 HUB5 English Evaluation Speech* (Katalognummer: LDC2002S09) (kurz: eval2000) wird vom Linguistic Data Consortium (LDC) bereitgestellt und enthält etwa 11 Stunden Material aus 40 Telefonkonversationen zwischen je zwei Personen. Je 20 Konversationen stammen aus den Switchboard und CALLHOME American English Speech Studien. Die Audio-Dateien sind im Sphere-Format und enthalten zwei Kanäle in µ-law[4] und 8000 Hz Samplingrate. Transkripte sind getrennt unter 2000 HUB5 English Evaluation Transcripts (Katalognummer LDC2002T43) erhältlich. [Ling 02] Die Verteilung von Nicht-Sprache, Sprache und Overlap wird in Tabelle 5.5 gezeigt:

Aufbereitung

[4] µ-law ist ein Digitalisierungsverfahren für analoge Audiosignale mit dem Ziel, ein besseres Signal-Rausch-Verhältnis zu erlangen.

Total	**Nicht-Sprache**	**Sprache**	1 Sprechender	2 Sprechende
99.9	33.5	66.4	55.2	11.2

Tabelle 5.5.: Verteilung von Nicht-Sprache und gleichzeitig Sprechenden im eval2000 Korpus in %. Rundungsfehler führen zu Total = 99.9%.

In Kaldi sind Rezepte zu finden, in denen mit eval2000 gearbeitet wurde. So kann das Skript[5] `eval2000_data_prep.sh` verwendet werden, um aus den Transkripten und Metadaten eine RTTM und weitere Dateien zu erzeugen. Zu beachten ist, dass diese Skripte Overlap vermeiden, das heißt, die RTTM-Datei wird 0 % Speaker Overlap aufweisen und die einzelnen Kanäle der Stereo-Audiodateien werden in je eine eigene wav-Datei extrahiert. Diese Herangehensweise resultiert aus der Zeit und den Zielen der urspünglichen Nutzung dieses Korpus. Overlap Detection wurde als besonders schwierig bewertet, weswegen Overlap zunächst nicht beachtet wurde, damit Diarization und ähnliche Systeme einfacher zu erforschen und zu vergleichen waren. Um eine RTTM-Datei mit Overlap zu erzeugen, wird die NIST Segment Time Mark (STM) - Datei verwendet. STM-Dateien enthalten das Transkript in folgender Form (siehe 5.6):

file	**channel**	**speaker**	**start**	**end**	**label**	**transcript**
en_4156	A	en_4156_A	301.85	302.48	<O,en,F,en-F>	OH YEAH
en_4156	A	en_4156_A	304.71	306.72	<O,en,F,en-F>	WELL I AM [...]
en_4156	A	en_4156_A	307.63	311.16	<O,en,F,en-F>	NO I AM [...]

Tabelle 5.6.: Die ersten drei Zeilen einer STM-Datei (letzte Spalte verkürzt dargestellt).

Mit dem Skript `stm2rttm.pl` aus den `kaldi/tools/`[6], lässt sich daraus eine RTTM-Datei erzeugen (siehe Tabelle 5.2), die Overlap enthält, allerdings nicht der Formatierung und Benennungs-Konvention der ursprünglichen RTTM-Datei entspricht. Gemäß der Dokumentation ist die Formatierung nicht notwendig und dient nur der Lesbarkeit. Der Channel wird trotz eigentlicher Stereoaufnahme mit 1 überschrieben, da beide Channels auf Mono gemixt werden mit `sox some_file.sph -t wav - remix 1,2`. Im Standardrezept `eval2000_data_prep.sh` werden beide Channels seperat herausgezogen und nicht gemixt.

```
awk '$1 ~ /SPEAKER/' \
  | awk '{
      $2=$2"-"$3;
      $3=1;
      $8=$2;
      $4=sprintf("%7.2f", $4);
      $5=sprintf("%7.2f", $5);
      print;
    }'
```

[5] kaldi/egs/fisher_swbd/s5/local/
[6] kaldi/tools/sctk-20159b5/bin/

```
10    | sort > rttm
```

Listing 5.2: Formatierung der ursprünglichen RTTM angleichen.

Eine zweite Möglichkeit ist, eine mitgelieferte Partitioned Evaluation Map (PEM) - Datei zu verwenden. Sie enthält Daten in Form von:

file	**channel**	**speaker**	**start**	**end**
en_4156	A	unknown_speaker	301.85	302.48
en_4156	A	unknown_speaker	304.71	306.72
en_4156	A	unknown_speaker	307.63	311.16

Tabelle 5.7.: Die ersten drei Zeilen einer PEM-Datei.

Leider konnten keine Dokumentation gefunden werden, die den Aufbau und gedachte Nutzung ausreichend erklärt. Insofern sollte die Nutzung als bedenklich eingestuft werden.

```
1  tail -n +3 hub5e_00.pem \
2      | awk '{
3          printf("%s %s %s %s %s %s %s %s %s\n",
4          "SPEAKER", $1"-"$2, "1", sprintf("%7.2f", $4),
5          sprintf("%7.2f", $5-$4), "<NA>", "<NA>", $1"-"$2, "<NA>");
6        }'
7      | sort > rttm
```

Listing 5.3: Konventierung und Formatierung der PEM-Datei zu einer RTTM-Datei.

5.4. Experimente

In diesem Kapitel werden die verarbeiteten Datensätze in die bestehenden Rezepte und Strukturen der Diarization Systeme der Baseline und EEND-EDA eingepflegt. Dazu wird das AMI-Datenset für Training und eval2000-Datenset für die Evaluierung genutzt.

5.4.1. Baseline

Als Features werden die ersten 40 MFCC genutzt, die mit 25ms Fenster-Größe, 10ms Schritt-Größe, 40 mel-Bins, $f_{min} = 20Hz$, $f_{max} = 8000Hz$ parametrisiert sind. Anschließend werden diese MFCCs mit einem Sliding Window von 300ms Länge Mittelwert normalisiert. Diese Features werden anschließend durch ein vortrainiertes Deep Neural Network (DNN) weiterverarbeitet, um X-Vektoren Embeddings zu gewinnen.

Das verwendete DNN heißt „CHIME 6 Baseline Diarization system“ [Snyd 18] und ist öffentlich auf Kaldi[7] zugänglich. Es wurde auf einer künstlich augmentierten Variante des Voxceleb Korpus vortrainiert. Als Vorbereitung für den Einsatz am DNN werden lange kontinuierliche Sprechsegmente in multiple 1,5 s lange Segmente unterteilt mit einer Schrittlänge von 10 s. Damit wird verhindert, dass nicht zu viele X-Vektoren aus einer Aufnahme benutzt werden. Alle Segmente unter 1,5 s werden für die weitere Betrachtung ausgeschlossen. Die X-Vektoren werden schlussendlich linear Whitening-transformiert und Mittelwert bereinigt.

Ein PLDA-Modell wird trainiert, indem die Ähnlichkeit der erstellten X-Vektoren paarweise berechnet wird. Dazu werden Sprechendeninformationen genutzt, um innerhalb und zwischen Klassen Varianzen zu berechnen, um Ähnlichkeiten der Features festzustellen. Bei der Umsetzung ist zu beachten, dass das Kaldi Standard Rezept zwar ein PLDA-Modell trainiert, aber nie zum Scoring verwendet. Hierfür muss das Skript `local/diarize_spectral.sh` angepasst werden, um statt Cosinus-Ähnlichkeit PLDA-Scores zu berechnen. Das passende Script heißt `score_plda.sh`. Es erwartet I-Vektoren, weshalb im Skript einige erwartete Dateinamen und Variablen auf X-Vektoren angepasst werden müssen.

Bei der Inferenz wird eval2000 bis zum PLDA-Scoring gleich verarbeitet. Die X-Vektoren werden über alle Paare von Segmenten mit dem trainierten PLDA-Modell gescored. Versuchsweise wird auch Cosinus-Scoring genutzt. Die Scorings werden anschließend durch Spectral Clustering geclustert. Die abschließend entstandene RTTM-Datei wird mit der Referenz verglichen. Das Ergebnis ist in der Tabelle 5.8 zu sehen. Wie erwartet, liegt FA bei 0,0, da mit Oracle SAD gearbeitet wurde. Ungewöhnlich ist, dass MS ebenfalls 0,0 ist.

In einem nachfolgenden Schritt wird Overlap Detection mit einem NN durchgeführt. Dazu wird die gleiche Feature-Konstellation wie zu Beginn des Kapitels genutzt, um ein TDNN LSTM mit Statistical Pooling 40 Epochen lang zu trainieren. Das fertige Modell erkennt dann bei der Inferenz Overlap-Regionen, die mit der Referenz verglichen werden. Aus diesem Ergebnis interessiert nur die FA, die auf das Ergebnis von zuvor aufaddiert wird. In diesem mehrschichtigen System übernimmt jeder Schritt eine spezialisierte Aufgabe und das Endergebnis ist die Kumulation aller Schritte.

[7] https://kaldi-asr.org/models/m12

	System				
Overlap	Similarity	MS	FA	SE	**DER**
no	cosine	0.0	0.0	45.2	45.2
	PLDA	0.0	0.0	33.8	33.8
yes	cosine	51.1	0.0	45.2	96.3
	PLDA	51.1	0.0	33.8	84.9

Tabelle 5.8.: Ergebnis der Baseline mit Oracle SAD und Spectral Clustering, PLDA oder Kosinusähnlichkeit und Overlap.

Der Baseline Ansatz benötigte für die vollständige Berechnung 40 Minuten.

5.4.2. EEND-EDA

Da in dieser Arbeit nicht die gleichen Voraussetzungen wie bei der Experimentbeschreibung von [Hori 20] gelten, wurden eine Reihe von Vereinfachungen durchgeführt.

Unterschiede

Das Team um Horiguchi et al. teilt den Datensatz auf und augmentiert diesen durch simulierte, fest definierte Overlap Segmente. Da Augmentation und Simulation zum einen den Rahmen der Arbeit etwas sprengen würde, muss zum anderen auf die Schwierigkeit der Durchführung jener mit dem AMI Datensatz hingewiesen werden. Genutzt wurde der Headset-Mix, der standardmäßig auf Mono runtergemixt ist. Da Sprechende nicht eindeutig entsprechend den Kanälen zugeordnet werden können und viele Störgeräusche wahrnehmbar sind, ist diskutabel, ob aus diesem Datenset simulierte Mixe von hoher Güte wären.

Daran schließt sich an, dass ebenfalls kein schrittweises adaptives Training gemacht werden kann. Horiguchi et al. beginnt mit dem Training von zwei Sprechenden und adaptiert das NN iterativ mit festen Anzahlen von Sprechenden. Dazu notwendig sind präzise konzipierte Datensets, die diese Arbeit nicht hat.

Durchführung

Für das Experiment werden als Feature-Set 345-dimensionale log-skalierte Mittelwert-korrigierte Mel-Spektrogramme mit 23 Mel-Bins (23-LogMel) für den Input an das SA-EEND verwendet. Das SA-EEND besteht aus vier gestapelten Transformer-Encoder. Nach der letzten Normalisierungsschicht wird eine Sequenz aus 256-dimensionalen Embeddings extrahiert. Das EDA errechnet aus den Embeddings Attractors. Der Grenzwert τ in 4.8 wurde auf 0,5 gesetzt und definiert, welche Attractors akzeptiert werden.

Für die Inferenz wird neben eval2000 auch das vordefinierte AMI-test Set genutzt. Die gesamte Rechenzeit betrug 40 Stunden.

Ergebnisse

Die Ergebnisse sind in Tabelle 5.9 aufgeführt. Bei beiden besten Modellen handelt es sich um die gemittelte Epoche 91 - 100. Die DER für eval2000 liegt bei 100 %, für das AMI-test set bei 261.21 %, wovon 255.8 % auf FA zurückzuführen ist.

System	Set	MS	FA	SE	DER
EEND-EDA	eval2000	0	100	0.0	100
	AMI-test	0	255.8	5.4	261.2

Tabelle 5.9.: Ergebnis der Inferenz von eval2000 und AMI-test auf dem trainierten EEND-EDA Modell in %.

Diskussion

Das Ergebnis ist ohne Zweifel sehr schlecht. Eine hohe FA kann bedeuten, dass häufig Sprache erkannt wurde, wo keine war. Daher ist es nicht verwunderlich, dass ein System, dass überall Sprache zu erkennen „meint“, keine Sprachsegmente verpasst und deswegen die MS 0 % ist. Verwunderlich ist, dass SE bei eval2000 0 % ist. Das heißt, keine einzige Sekunde wurde einem falschen Sprecher zugewiesen, was wiederum darauf hindeuten würde, dass das System zuverlässig, wenn auch „überschwänglich“, die Identität von Sprechenden erkannt haben könnte.

Nicht auszuschließen sind technische und handwerkliche Fehler. Da dies ein erstes Projekt in diesem Fachgebiet darstellt, kann es trotz aller Sorgfalt zu Unzulänglichkeiten und Unachtsamkeiten gekommen sein. Kaldi ist ein sehr mächtiges Framework mit vielen Skripten und umfangreichen Bibliotheken, sodass nicht immer sofort eingängig ist, welche Eingabedaten zu welchen Programmen nötig und welche (Zwischen-)Ergebnisse zu erwarten sind.

Ein weiterer offener Diskussionspunkt ist die Domänen-Ungleichheit zwischen dem Trainings- und dem Evaluationsdatenset. Die AMI-Trainingsdaten sind aus Headset-Mikrofonaufnahmen bei einem Meeting entstanden, währenddessen eval2000-Daten bei Telefonkonversationen entstanden sind. Da Teams zum Teil Domänen-spezialisierte Sub-Systeme entwickeln (vergleiche [Wang 21b]), liegt die Vermutung nahe, dass Domäneninformationen ebenfalls eine wichtige Rolle für erfolgreiche Diarization spielen, die scheinbar nicht zu unterschätzen ist. Da allerdings mit dem AMI-test Datensatz eine passende Domäne getestet wurde und trotzdem sehr schlechte Ergebnisse erzielt wurden, muss von einem zugrunde liegenden Problem, wie technische oder handwerkliche Fehler ausgegangen werden.

Kapitel 6.

Zusammenfassung

Diese Arbeit sollte ein Diarization System für Arzt-Patienten-Gespräche vorbereiten, das in der Lage sein soll, mit Overlap von zwei und mehr Sprechenden umzugehen. Dazu wurden die Grundlagen von ML und Sprache erarbeitet und Methoden und Modelle zur Erstellung und Verarbeitung von hergeleiteten Features besprochen. Es wurden Qualitätskriterien aufgestellt, nach denen geeignete aktuelle Diarization Systeme bewertet wurden. Die Analyse der besten drei Platzierten der DIHARD Challenge III zeigte auf, dass als Testsystem das Subsystem SA-EEND-EDA aus dem Hitachi-JHU-System am geeignetsten ist. Als Baseline wurde ein Standard Kaldi-Rezept verwendet, um Vergleichbarkeit und Reproduzierbarkeit zu gewährleisten. Die Baseline trainiert ein PLDA-Modell mit X-Vektoren und nutzt Spectral Clustering zum clustern von PLDA-gescorten X-Vektor Embeddings. Für Overlap Detection wurde ein TDNN-LSTM trainiert. Als Trainingsdatenset wurde AMI und als Evaluationsset eval2000 vorgegeben. Die Baseline erreichte 96,3 %, SA-EEND-EDA-System 100 % DER auf dem eval2000 Datensatz. Beide Systeme schnitten schlecht ab. Die Gründe hierfür können vielfältig sein. Beispielsweise wäre eine naheliegende Fehlerquelle, die das Versagen beider Modelle erklären könnte, die fehlerhafte Nutzung und Verarbeitung des Trainings- und/oder Evaluierungs-Datensatzes. Das Debugging unter Linux mit `bash` stellte sich als schwierig heraus. Es wurde keine beweisbare Erklärung für das schlechte Abschneiden der Systeme gefunden.

6.1. Ausblick

Für weiterführende Arbeiten sind zahlreiche Ansatzpunkte für Verbesserung und Erweiterungen denkbar:

1. Auffinden und Behebung des zugrunde liegenden Fehlers.
2. Erweiterung des Trainingsdatensatzes um weitere Korpora zur gezielten Abdeckung der geforderten Domäne in Bezug auf Arzt-Patienten-Gespräche und zur Abdeckung mehrerer verschiedener Sprechenden in jeweiligen Aufnahmen (beispielsweise dedizierte Aufnahmen von Gesprächen zu zweit, dritt, viert).

3. Einbindung des eigentlichen Korpus von Gesprächen zwischen Arzt und Patient.
4. Datenaugmentation und simulierte Mixe können den Basisdatensatz qualitativ und quantitativ erweitern. Diese sollten eine Ergänzung zu 2 sein.
5. Mit größerer Datenvielfalt durch Punkte 2 und 4 kann auch die Adaption in EEND-EDA vollzogen werden.
6. SAD Nutzung bei der Baseline, um einen fairen Vergleich zu Systemen zu schaffen, die implizite SAD machen.
7. Es können andere Baselines in Erwägung gezogen werden. Besonders erwähnenswert sind Baselines, die bei größeren Challenges wie DIHARD oder Ähnlichem genutzt wurden. Diese ermöglichen den direkten Vergleich mit der aktuellen Forschung.
8. Zwischenzeitlich sind Erweiterungen des EEND-EDA Systems veröffentlicht worden, wie zum Beispiel EEND-EDA with Speaker-Tracing Buffer for Flexible Numbers of Speakers (FLEX-STB) [Xue 21], die sogar Echtzeitfähigkeit versprechen.
9. Durch Implementierung und Testung mehrerer verschiedener Testsysteme erhöht man die Chance auf ein passendes System für das eigentliche Zieldatenset.
10. Visualisierung von EEND-EDA. Da dieses System sehr abstrakt ist, könnte eine Visualisierung der Mechanismen helfen, ein besseres Verständnis zu schaffen.
11. Wünschenswert wäre auch ein softwaretechnischer Umbau und dokumentarischer Ausbau von Rezepten. Es gibt einige Bereiche in Rezepten, die Ergebnisse berechnen, die niemals verwendet werden. Möchte man ein Paper oder Rezepte nachvollziehen, so kann es schnell für Verwirrung sorgen.

6.2. Fazit

Das Forschungsgebiet der ASR und Speaker Diarization ist hochaktuell und sehr aktiv. Gefühlt wochenweise werden neue Diarization Systeme oder Erweiterungen und Verbesserungen publiziert. Es herrscht großer Wettbewerb und es gibt viele Möglichkeiten teilzuhaben. Durch all die Hektik und Fluktuation scheint es an einem roten Faden zu fehlen, an dem sich Anfänger strukturiert durch die Geschichte der Speaker Diarization arbeiten und beispielsweise Kaldi kennenlernen können. Erschwerend kommt hinzu, dass viele Paper keine Repositorys angeben, obwohl so die Methoden und Ergebnisse reproduzierbar und verifizierbar gemacht werden könnten.

Die schlechten Ergebnisse der genutzten Systeme implizieren, dass die geforderten Ziele ungenügend erreicht worden sind. Im aktuellen Zustand vom Diarizaton System SA-EEND-EDA in diesem Projekt ist keine nutzbare Diarization mit Overlap zu erwarten. Die erwähnten Ideen zu Verbesserungen und Erweiterungen könnten die Qualtiät signifikant steigern, um das Refernzsystem nutzbar zu machen.

Anhang A.

Klinger-Fork: Github EEND-EDA

In dieser Arbeit wurde als Codebasis https://github.com/hitachi-speech/EEND verwendet. Für die Bearbeitung dieser Arbeit wurde ein Fork https://github.com/nusoos/EEND genutzt. Wenn nicht anderes gekennzeichnet, sind die Urheber Hitachi et al. oder entsprechende Autoren der Dateien auf Kaldi.

Abbildungsverzeichnis

Tabellenverzeichnis

List of Listings

Literaturverzeichnis

[Amaz] Amazon. "Amazon Personalize — Beschleunigen Sie die Erstellung personalisierter Benutzererfahrungen in Echtzeit". https://aws.amazon.com/de/personalize/.

[Angu 06] X. Anguera. *Robust Speaker Diarization for meetings*. PhD thesis, Barcelona, Spain, 2006.

[Bru14] M. Bruß. "Grundlagen der Psychoakustik, Teil 1". 2014. https://www.fairaudio.de/hintergrund/psycho-akustik-artikel-1-dwt/psycho-akustik-artikel-3-dwt/.

[Chan 74] B. Chandrasekaran and A. Jain. "Quantization Complexity and Independent Measurements". *IEEE Transactions on Computers*, Vol. C-23, No. 1, pp. 102–106, 1974.

[Eule 06] S. Euler. *Grundkurs Spracherkennung*. Vieweg, 2006.

[Fisc 06] J. Fiscus, J. Ajot, M. Michel, and J. Garofolo. "The Rich Transcription 2006 Spring Meeting Recognition Evaluation". Rich Transcription Spring Meeting Recognition Evaluation 2006, 2006-05-04 2006.

[Fuji 19a] Y. Fujita, N. Kanda, S. Horiguchi, K. Nagamatsu, and S. Watanabe. "End-to-End Neural Speaker Diarization with Permutation-Free Objectives". 2019.

[Fuji 19b] Y. Fujita, N. Kanda, S. Horiguchi, Y. Xue, K. Nagamatsu, and S. Watanabe. "End-to-End Neural Speaker Diarization with Self-Attention". In: *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 296–303, 2019.

[Henn 08] H. Henn, G. R. Sinambari, and M. Fallen. *Physiologische Grundlagen des Hörens und objektive Lautstärke*, pp. 183–214. Vieweg+Teubner, Wiesbaden, 2008.

[Hori 20] S. Horiguchi, Y. Fujita, S. Watanabe, Y. Xue, and K. Nagamatsu. "End-to-End Speaker Diarization for an Unknown Number of Speakers with Encoder-Decoder Based Attractors". 2020.

[Hori 21] S. Horiguchi, N. Yalta, P. Garcia, Y. Takashima, Y. Xue, D. Raj, Z. Huang, Y. Fujita, S. Watanabe, and S. Khudanpur. "The Hitachi-JHU DIHARD III System: Competitive End-to-End Neural Diarization and X-Vector Clustering Systems Combined by DOVER-Lap". 2021.

[Jane 91] S. Janet. "NIST SPHERE Format Documentation". 1991. https://isip.piconepress.com/projects/speech/software/tutorials/production/fundamentals/v1.0/section_02/text/nist_sphere.text.

[Klei 17] A. Klein. "Hard Drive Cost Per Gigabyte". 2017. https://www.backblaze.com/blog/hard-drive-cost-per-gigabyte/.

[Komo 14] M. Komorowski. "a history of storage cost (update)". 2014. http://www.mkomo.com/cost-per-gigabyte-update.

[Ling 02] Linguistic Data Consortium. "2000 HUB5 English Evaluation Speech LDC2002S09". 2002. https://catalog.ldc.upenn.edu/LDC2002S09.

[Mcco 05] I. Mccowan, J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska Masson, W. Post, D. Reidsma, and P. Wellner. "The AMI meeting corpus". *Int'l. Conf. on Methods and Techniques in Behavioral Research*, 01 2005.

[Mell 18] R. F. de Mello and M. A. Ponti. *Machine Learning - A Practical Approach on the Statistical*. Springer, 2018.

[NIST 09] NIST. "The 2009 (RT-09) Rich Transcription Meeting Recognition Evaluation Plan". 2009. https://web.archive.org/web/20170119114252/http://www.itl.nist.gov/iad/mig/tests/rt/2009/docs/rt09-meeting-eval-plan-v2.pdf.

[Park 20] T. J. Park, K. J. Han, M. Kumar, and S. Narayanan. "Auto-Tuning Spectral Clustering for Speaker Diarization Using Normalized Maximum Eigengap". *IEEE Signal Processing Letters*, Vol. 27, p. 381–385, 2020.

[Pfis 08] B. Pfister and T. Kaufmann. *Sprachverarbeitung: Grundlagen Und Methoden Der Sprachsynthese Und Spracherkennung (Springer-Lehrbuch)*. Springer Publishing Company, Incorporated, 1 Ed., 2008.

[Pove 11] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely. "The Kaldi Speech Recognition Toolkit". In: *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, IEEE Signal Processing Society, Dec. 2011. IEEE Catalog No.: CFP11SRW-USB.

[Rena 07] S. Renals, T. Hain, and H. Bourlard. "Recognition and understanding of meetings the AMI and AMIDA projects". *2007 IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2007, Proceedings*, 01 2007.

[Rouv 15] M. Rouvier, P.-M. Bousquet, and B. Favre. "Speaker diarization through speaker embeddings". In: *2015 23rd European Signal Processing Conference (EUSIPCO)*, pp. 2082–2086, 2015.

[Ryan 21] N. Ryant, P. Singh, V. Krishnamohan, R. Varma, K. Church, C. Cieri, J. Du, S. Ganapathy, and M. Liberman. "The Third DIHARD Diarization Challenge". 2021.

[Sahi 12] M. Sahidullah and G. Saha. "Comparison of Speech Activity Detection Techniques for Speaker Recognition". 2012.

[Silv 16] D. Silver and D. Hassabis. "AlphaGo: Mastering the ancient game of Go with Machine Learning". 2016. https://ai.googleblog.com/2016/01/alphago-mastering-ancient-game-of-go.html.

[Snyd 18] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur. "X-vectors: Robust DNN Embeddings for Speaker Recognition". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2018. http://www.danielpovey.com/files/2018_icassp_xvectors.pdf.

[Stad 09] T. Stadelmann and B. Freisleben. "Unfolding speaker clustering potential: A biomimetic approach". pp. 185–194, 01 2009.

[Stev 37] S. S. Stevens, J. E. Volkmann, and E. B. Newman. "A Scale for the Measurement of the Psychological Magnitude Pitch". *Journal of the Acoustical Society of America*, Vol. 8, pp. 185–190, 1937.

[Wang 21a] W. Wang, Q. Lin, D. Cai, L. Yang, and M. Li. "The DKU-Duke-Lenovo System Description for the Third DIHARD Speech Diarization Challenge". 2021.

[Wang 21b] Y. Wang, M. He, S. Niu, L. Sun, T. Gao, X. Fang, J. Pan, J. Du, and C.-H. Lee. "USTC-NELSLIP System Description for DIHARD-III Challenge". 2021.

[Wojc 20] S. Wojcicki. "YouTube at 15: My personal journey and the road ahead". 2020. https://blog.youtube/news-and-events/youtube-at-15-my-personal-journey/.

[Xue 21] Y. Xue, S. Horiguchi, Y. Fujita, Y. Takashima, S. Watanabe, P. Garcia, and K. Nagamatsu. "Online Streaming End-to-End Neural Diarization Handling Overlapping Speech and Flexible Numbers of Speakers". 2021.

[Yu 17] D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen. “Permutation Invariant Training of Deep Models for Speaker-Independent Multi-talker Speech Separation”. 2017.

Glossar

AHC Agglomerative Hierarchical Clustering. i, 19, 21

AMI Augmented Multi-Party Interaction. i, viii, 2, 31

ASR Automatic Speech Recognition. i, 1, 7

DCT Discrete Cosinus-Transformation. i, 12

DER Diarization Error Rate. i, v, 2

DKU Duke Kunshan University. i, 21

DNN Deep Neural Network. i, 35, 36

DOVER Diarization Output Voting Error Reduction. i, 20

EDA Encoder-Decoder-Attractor. i, vii, 23, 24, 26

EEND End-to-end neural network-based speaker diarization model. i, vii, 23, 24, 26

eval2000 2000 HUB5 English Evaluation Speech. i, viii, 2, 33

HMM Hidden Markov Model. i, 19

ISS Iterative Speech Seperation. i, 19

JHU John Hopkins University. i, 20

LDA Linear Discriminant Analysis. i

LDC Linguistic Data Consortium. i, 28, 33

MFCC Mel Frequency Cepstral Coefficients. i, 12

ML Machine Learning. i, 5

NELSLIP National Engineering Laboratory for Speech and Language Information Processing. i, 19

NIST National Institute of Standards and Technology. i, 28

NN Neuronale Netze. i, 13

PEM Partitioned Evaluation Map. i, 35

PIT Permutation Invariant Training. i, 24

PLDA Probabilistic Linear Discriminant Analysis. i, 19

RTTM Rich Transcription Time Marked. i, 28

SA Self-Attentive. i, 23

SAD Speech Activity Dettection. i, 12, 13

SPHERE Speech Header Resources. i, 28

STFT Short-Time Fourier Transformation. i, 9

STM Segment Time Mark. i, 34

TDNN Time-Delay Neural Network. i, 20

TS Target-Speaker. i, 19

USTC University of Science and Technology of China. i, 19

VAD Voice Activity Detection. i, 12, 19

VB Variational Bayes. i, 19

VBx VB-HMM x-vectors. i, 20