Zürcher Hochschule
für Angewandte Wissenschaften

School of
Engineering

InIT Institut für angewandte
Informationstechnologie

Projektarbeit (Informatik)

Automatische Transkription von Interviews

Autoren	Nicolas Elvis Eckhart Mathias Richard Marxer Malgorzata Ulasik
Hauptbetreuung	Mark Cieliebak

Datum	18.01.2019

Zürcher Hochschule für Angewandte Wissenschaften

School of Engineering

Erklärung betreffend das selbständige Verfassen einer Projektarbeit an der School of Engineering

Mit der Abgabe dieser Projektarbeit versichert der/die Studierende, dass er/sie die Arbeit selbständig und ohne fremde Hilfe verfasst hat. (Bei Gruppenarbeiten gelten die Leistungen der übrigen Gruppenmitglieder nicht als fremde Hilfe.)

Der/die unterzeichnende Studierende erklärt, dass alle zitierten Quellen (auch Internetseiten) im Text oder Anhang korrekt nachgewiesen sind, d.h. dass die Projektarbeit keine Plagiate enthält, also keine Teile, die teilweise oder vollständig aus einem fremden Text oder einer fremden Arbeit unter Vorgabe der eigenen Urheberschaft bzw. ohne Quellenangabe übernommen worden sind.

Bei Verfehlungen aller Art treten die Paragraphen 39 und 40 (Unredlichkeit und Verfahren bei Unredlichkeit) der ZHAW Prüfungsordnung sowie die Bestimmungen der Disziplinarmassnahmen der Hochschulordnung in Kraft.

Ort, Datum:

……

Unterschriften:

……

……...

……

Zusammenfassung

Die Transkription gesprochener Konversationen ist ein Use Case, der in duzenden Gebieten wie der Politik, der Medizin oder der Geschäftswelt vorhanden ist. Seien es Meetings, Interviews im Medienumfeld oder Bewerbungsgespräche, die meisten Unternehmen berufen sich auf menschliche Transkription, was nicht nur ein kosten-, sondern auch ein zeitaufwändiger Prozess ist. Mehr und mehr zeigt sich jedoch automatisierte Audio-Transkription als realistische Alternative. Die Auswahl an Programmen, die ohne technische Kenntnisse auf der Seite des Benutzers zur Audio-Transkription verwendet werden können, ist jedoch klein und viele wichtige Funktionen, wie Sprecher- oder Pausenerkennung, werden nicht unterstützt.

Das Ziel dieser Arbeit war es, ein System für die automatische Transkription von Interviews zu entwickeln. Es sollten Audio-Dateien der Anwendung übergeben werden können und das Transkript zurückerhalten werden. Zusätzlich sollte das Audio abgespielt und die Transkription exportiert werden können.
Zuerst wurde eine Evaluation von bestehenden Tools und das Aufarbeiten der theoretischen Grundlagen durchgeführt, anschliessend Anforderungen definiert und das modulare Design entworfen. Im Anschluss wurde die Anwendung mittels einem Python-Backend, einer Electron Anwendung mit Vue.js als Frontend-Framework und der Google-Cloud API für die Transkription umgesetzt. Um die Leistungsfähigkeit der Anwendung zu überprüfen wurde im Anschluss eine automatisierte Evaluation mit insgesamt 108 Audio-Dateien durchgeführt.

Der Prototyp mit den wichtigsten Use Cases wurde implementiert, jedoch bleiben viele Erweiterungsmöglichkeiten wünschenswert. So könnten zum Beispiel andere Sprachen, wie Deutsch oder sogar Schweizerdeutsch unterstützt werden oder die Anwendung für mobile Geräte portiert werden.

Inhaltsverzeichnis

1 Einleitung

1.1 Ausgangslage

Die Transkription gesprochener Konversationen ist ein Use Case, der in duzenden Gebieten wie der Politik, der Medizin oder der Geschäftswelt vorhanden ist. Seien es Meetings, Interviews im Medienumfeld oder Bewerbungsgespräche, die meisten Unternehmen berufen sich auf menschliche Transkription, was nicht nur ein kosten-, sondern auch ein zeitaufwändiger Prozess ist. Mehr und mehr zeigt sich jedoch automatisierte Audio-Transkription als realistische Alternative. Mit Weiterentwicklungen in Feldern wie Natural Language Processing, Audio-Analyse und dem maschinellen Lernen konnten Tech-Konzerne wie IBM, Google und Microsoft Speech-to-Text Dienste mit hoher Genauigkeit und tiefen Preisen der Öffentlichkeit zugänglich machen.

Während diese Dienste sich gut für Enterprise-Lösungen eignen, sind sie für Endbenutzer nicht optimal. Die Auswahl an Programmen, die ohne technische Kenntnisse auf der Seite des Benutzers zur Audio-Transkription verwendet werden können, ist klein und viele wichtige Funktionen, wie Sprecher- oder Pausenerkennung, werden nicht unterstützt.

1.2 Zielsetzung

Das Ziel dieser Arbeit ist es, ein komplettes System für die Interview Transkription zu entwickeln. Dabei soll eine Audio-Datei der Anwendung übergeben werden können und die Transkription zurückbekommen und angezeigt werden. Um dieses Ziel zu erreichen sollen zuerst die theoretischen Grundlagen aufgearbeitet und eine Analyse bestehender Tools durchgeführt werden. Anschliessend kann, darauf aufbauend, die Anwendung spezifiziert und entwickelt werden. Im letzten Schritt soll die implementierte Anwendung evaluiert werden.

1.2.1 Spezifikation

Damit die implementierte Anwendung gut bedienbar ist und die Ziele erreicht, muss die Anwendung einfach zu installieren und zu bedienen sein. Damit dies erreicht wird, muss eine Audio-Datei über eine einfache grafische Benutzerschnittstelle in die Anwendung geladen werden können, anschliessend transkribiert und angezeigt werden. Dabei muss erkennbar sein, welche Abschnitte in der Transkription von welchen Sprechern stammen. Ausserdem sollen Textpassagen direkt abgespielt werden können, um Fehler einfach zu entdecken. Die Anwendung soll auch die Möglichkeit besitzen, eine Transkription einfach zu exportieren und das gesamte Projekt zu speichern, um beliebig daran arbeiten zu können. Zuletzt ist ein wichtiger Aspekt, dass der verwendete Transkriptions-Dienst möglichst einfach ersetzbar sein soll.

1.2.2 Aufbau der Arbeit

Zuerst werden die theoretischen Grundlagen zu Spracherkennung und Sprechertrennung beschrieben. Anschliessend werden bereits bestehende Transkriptions-Tools für Benutzer sowohl Entwickler evaluiert und beschrieben. Danach wird das Vorgehen der Entwicklung anhand Anforderungen und Design erklärt. Im Anschluss wird eine Evaluation mit der implementierten Anwendung durchgeführt, um die Leistungsfähigkeit festzustellen. Abschliessend wird der Projektverlauf diskutiert, Entscheidungen hinterfragt und einen Ausblick auf mögliche Erweiterungen gewagt.

1.2.3 Terminologie

Auslassung, Deletion (in Bezug auf WER)	Ein fehlerhaft ausgelassenes Wort in einem Transkript.
Äusserung, Utterance, Sprecherabschnitt	Aussage beliebiger Länge eines Sprechers die beim nächsten Sprecherwechsel endet.
Diarisierung, Diarization, Sprecher Diarization, Speaker Diarization	Erkennen und unterscheiden von verschiedenen, unbekannten Sprechern in einer Audio-Datei.
Einfügung, Insertion (in Bezug auf WER)	Ein fehlerhaft eingefügtes Wort in einem Transkript.
Ersetzung, Substitution (in Bezug auf WER)	Ein falsch transkribiertes Wort in einem Transkript.
Intersprechervariabilität	Unterschiede in der Aussprache (Dialekt, Betonung, etc.) zwischen verschiedenen Sprechern.
Intrasprechervariabilität	Unterschiede in der Aussprache eines Wortes desselben Sprechers
Korpus	Menge an Audio-Dateien mit Äusserungen von verschiedenen Sprechern die zu Testzwecken verwendet wird.
Merkmal, Feature (in Bezug auf Audio)	Verschiedene Eigenschaften, die dazu dienen, Audio zu charakterisieren.
Nulldurchgangsrate, Zero Crossing Rate	Feature, welches wenn tief auf Sprachsegmente und wenn hoch auf Ruhe hinweist.
Speech-Korpus	Korpus von Audio-Dateien die zur Evaluation der Interview Transkription Applikation verwendet wurden.
Spektralbereich / Frequenzspektrum	Kontinuierliches Spektrum das mittels einer Fourier-Transformation auf einem Audio-Signal berechnet wird.
Spracherkennung, Speech-to-Text (STT), (Automatic) Speech Recognition (ASR)	Die gesprochene Sprache in einer Audio-Datei zu Text transkribieren.
Sprachmodell	Regelsystem welches die Struktur einer Sprache beschreibt.
Sprecher-Identifikation	Audio-Datei dem korrekten Sprecher aus einer Menge von Sprecher-Modellen zuweisen.
Sprechertrennung	Überbegriff für Sprecher-Verifikation, Sprecher-Identifikation und Sprecher Diarization.
Sprecher-Verifikation	Prozess der Verifikation ob ein Audio-Segment zu einem vorhandenen Sprecher-Modell passt.

Word Recognition Rate (WR), Worterkennungsrate	Metrik welche prozentual die Anzahl korrekt transkribierten Wörter angibt.
Word-Error-Rate (WER), Wortfehlerrate	Metrik welche die Fehlerrate einer Transkription beschreibt.
Worterkennung	Anhand der Charakteristiken eines Audio-Segmentes das darin gesprochene Wort erkennen.
Zeitsignal	Verlaufs eines Audio-Signals als Funktion der Zeit.

2 Theoretische Grundlagen

Eine maschinelle Transkription von einem Interview erfordert einen Einsatz von zwei Technologien: Spracherkennung, damit das Interview in eine textliche Form umgewandelt werden kann und Sprecher Diarization, damit die verschiedenen, unbekannten Sprecher, die an dem Interview teilnehmen, unterschieden werden können. Diese zwei Konzepte liegen der in dieser Arbeit beschriebenen Transkriptions-Anwendung zugrunde und werden im Folgenden detailliert dargestellt. Somit wird ein theoretischer Hintergrund für die später beschriebene Implementierung und Evaluation der Anwendung geliefert.

2.1 Spracherkennung

Unter Spracherkennung (Speech Recognition) wird das Umsetzen eines Sprachsignals in eine textuelle Form verstanden. In der Literatur werden neben Spracherkennung und Speech Recognition auch die Begriffe Automatic Speech Recognition (ASR) und Speech-to-Text (STT) verwendet.

Es ist ein interdisziplinäres Gebiet, dass sich auf Wissen und Methoden mehrerer Disziplinen, wie Linguistik, Akustik, Signalverarbeitung, Statistik und Informatik, stützt [3]. In den nachfolgenden Kapiteln werden die theoretischen Grundlagen der Spracherkennung kurz beschrieben.

2.1.1 Sprachstruktur

Bei der Beschreibung der Struktur der Sprache unterscheidet die Linguistik verschiedene Ebenen. Für die Spracherkennung hat die Beschreibung auf der phonemischen Ebene den Vorrang. Aus phonemischer Sicht betrachtet, stellen die Lauten (Phone) die kleinsten Einheiten der Sprache dar. Lautsprache ist somit eine Abfolge von Phonen, die in Form eines Sprachsignals physikalisch erfasst werden. Die Laute in dem Sprachsignal sind allerdings nicht scharf voneinander abgegrenzt, weder in zeitlicher Hinsicht, noch bezüglich der charakteristischen Eigenschaften. Die phonetischen Eigenschaften des Sprachsignals verändern sich kontinuierlich von einem Laut zum nächsten, auch über die Wortgrenzen hinweg [3]. Daher ist eine eins-zu-eins-Abbildung der aufgenommenen Sprache auf vorher gespeicherte Signale von bekannten Worten nicht realisierbar. Um eine Worterkennung zu ermöglichen, müssen daher verschiedene Mittel eingesetzt werden. Dazu zählen Merkmalgewinnung und statistische Modelle für die Worterkennung.

2.1.2 Merkmalgewinnung

Das Sprachsignal entsteht als Resultat der Umwandlung der vom Menschen produzierten Schallwellen in zeitabhängige, elektrische Signale. Es liegt daher vorerst in einer analogen Form vor. Zur Verarbeitung von Sprachsignalen und zur Merkmalgewinnung werden allerdings die Mittel der digitalen Signalverarbeitung eingesetzt. Daher muss ein analoges Signal in ein digitales umgewandelt werden. Dazu werden die Techniken für Abtastung und Quantisierung eingesetzt [3]. Das Sprachsignal wird an einer endlichen Zahl äquidistanter Stützstellen abgetastet. Die resultierende Folge von Abtastwerten wird anschliessend quantisiert. Die Quantisierung besteht in Annäherung von jedem Abtastwert durch den Repräsentanten seiner zugehörigen Quantisierungsstufe (vgl. Abbildung 1) [22].

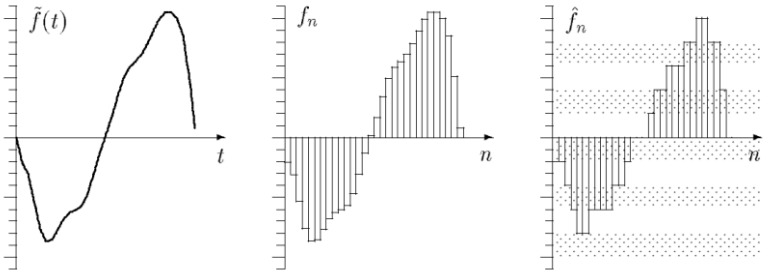


Abbildung 1: Abtastung und Quantisierung des Sprachsignals (1. Sprachsignal in analoger Form, 2. Sprachsignal als Folge von Abtastwerten, 3. Quantisierungsstufen vom Sprachsignal) [22]

Ein so transformiertes, digitalisiertes Sprachsignal ist eine Folge von diskreten Werten. Im nächsten Schritt der Vorverarbeitung wird eine Filterung vorgenommen. Filterung ist die Unterscheidung von Umgebungsgeräuschen und Sprache. Dazu kann zum Beispiel die Energie des Signals oder die Nulldurchgangsrate herangezogen werden. Eine hohe Nulldurchgangsrate ist ein Zeichen für stimmlose Laute, während eine niedrige Nulldurchgangsrate ein Zeichen für stimmhafte Laute ist. Zusammen mit der Energie des Signals kann auf diese Art und Weise eine Unterscheidung zwischen Sprache und Pause oder Sprache und Geräuschen getroffen werden [3].

Für die Spracherkennung ist nicht nur das Zeitsignal (Beschreibung des Signalverlaufs als Funktion der Zeit), sondern vorwiegend das Signal im Frequenzbereich relevant. Es wird daher eine Transformation des Zeitsignals in den Spektralbereich durchgeführt. Das Signal wird in kurze, zeitliche, 5-30ms andauernde Segmente (Frames) einteilt, für die eine Fourier-Transformation durchgeführt wird. Als Resultat der Transformation werden Kurzzeit-Frequenzspektren des Signals für jeden Frame erstellt. Anschliessend werden die Frames auf ihre spektralen oder periodischen Eigenschaften hin analysiert. Es werden zum Beispiel Cepstrum-Koeffizienten, Energiewerte oder Frequenzbereichsstärken berechnet [3,4,22,23].

Da bei sprachlicher Produktion eines Wortes die Laute mit variabler Dauer und in unterschiedlicher spektraler Zusammensetzung realisiert werden, wird für jeden Frame eine nicht vorhersagbare Anzahl von Merkmalvektoren erstellt. Jeder Vektor umfasst neben seinem phonetischen Gehalt auch sprecher- und umgebungsbedingte Informationsanteile. Der nächste Schritt der Spracherkennung besteht folglich darin, aus den berechneten Merkmalvektoren die wahrscheinlichsten Wörter zu erschliessen.

2.1.3 Worterkennung

Für die Worterkennung werden Hidden-Markov-Modelle (HMM), Deep Neural Networks (DNN) oder Hybridmodelle (aus Neuronalen und HMM-Anteilen) verwendet. Dabei stellen Hidden Markov Modelle eine Technik dar, die weitaus grösste Verbreitung findet und daher wird sich das Kapitel auf die Darstellung von diesem Ansatz konzentrieren.

Unter einem Hidden-Markov-Modell (HMM) versteht man zwei gekoppelte Zufallsprozesse. Der erste ist ein Markov-Prozess mit einer Anzahl Zuständen, die als S_1, ..., S_N bezeichnet werden. Sie steuern den zweiten Zufallsprozess. Dieser erzeugt zu jedem (diskreten) Zeitpunkt t gemäss einer zustandsabhängigen Wahrscheinlichkeitsverteilung eine Beobachtung x_t. Beim Durchlaufen einer Sequenz von Zuständen $Q = q_1 q_2 ... q_T$, mit $q_i \in \{S1, S2, ..., SN\}$, erzeugt das HMM eine Sequenz von Beobachtungen $X = x_1 x_2 ... x_T$. Ein HMM $\boldsymbol{\lambda}$ wird durch zwei Grössen beschrieben. Erstens

sind das die Zustandsübergangswahrscheinlichkeiten $a_{ij} = P(q_t=S_j \mid q_{t-1}=S_i)$ und zweitens die Beobachtungswahrscheinlichkeiten $b_j(x)= P(x|S_j)$: $\boldsymbol{\lambda} = (A, B)$.
A stellt dabei die N×N-Matrix der Zustandsübergangswahrscheinlichkeiten dar und B umfasst die Beobachtungswahrscheinlichkeitsverteilungen in den emittierenden Zuständen S_2, ..., S_{N-1} [3].

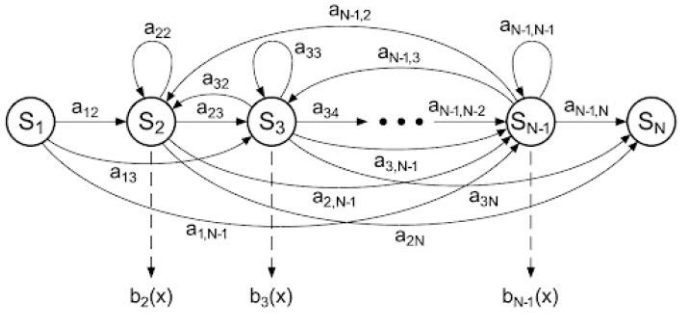


Abbildung 2: HMM mit N Zuständen [3]

Im Kontext der automatischen Spracherkennung handelt es sich bei HMM um Beantwortung der Frage, welche Zustandsfolge den Beobachtungen am wahrscheinlichsten zugrunde liegt. Dabei ist die, als Ergebnis der Signalanalyse berechnete Folge von Merkmalsvektoren, als Menge von Beobachtungen zu betrachten und die Laute werden als Zustände in dem Modell verstanden. Die Aufgabe der Spracherkennung besteht somit in der Dekodierung der tatsächlich geäusserten Lautfolge bei gegebenen Merkmalsvektoren. Dazu greift man auf diejenige Lautfolge zurück, die mit Hilfe von bayesischen Formel die grösste Wahrscheinlichkeit hat. Hat man die wahrscheinlichsten Lautfolgen, so lassen sich über das Sprachmodell die wahrscheinlichsten Wörter finden [24].

2.1.4 Sprachmodell

Mit dem Sprachmodell wird ein Regelsystem gemeint, das die Struktur der Sprache beschreibt. Durch die Sprachmodellierung kann einer Wortfolge W eine a priori Wahrscheinlichkeit P(W) zugeordnet werden. Die Aufgabe des Sprachmodells besteht darin, die Wortfolgen in Texten linguistisch zu bewerten, die Wahrscheinlichkeit bestimmter Wortkombinationen zu bestimmen und dadurch falsche oder unwahrscheinliche Hypothesen auszuschliessen, den Such-Raum für mögliche folgende Wörter einzuschränken und die Wahrscheinlichkeit der Hypothesen in Kombination mit dem akustischen Modell zu berechnen [25]. Das Sprachmodell stellt somit eine wichtige Wissensquelle dar, die zur Steigerung der Erkennungsleistung für Spracherkennung genutzt wird.

2.1.5 Herausforderungen bei der Spracherkennung

Auf den Sprechprozess wirken sich mehrere Faktoren. So können beispielsweise der Dialekt, die Sprechgewohnheiten, die Physiologie des Vokaltraktes, aber auch der momentane emotionale Zustand des Sprechers Einfluss haben. Auch Umgebungsgeräusche können auf den Sprechprozess wirken und so die Resultate der Spracherkennung beeinflussen. In der Regel hat auch die Übertragung der Schallwellen (die Raumakustik) bzw. des elektrischen Signals (suboptimale Komponenten des Aufnahmesystems, Mikrofoncharakteristik, Signalcodierung und -kompression, etc.) einen Einfluss auf das schlussendlich vorhandene Sprachsignal. Der Einfluss äußert sich einerseits in Dämpfungen oder Verstärkungen von einzelnen Frequenzen oder ganzen Frequenzbereichen, was die Aufgabe der Spracherkennung beträchtlich erschwert.

Dazu kommt das Problem der Intrasprechervariabilität, der Erscheinung, dass derselbe Sprecher dasselbe Wort immer wieder auf eine andere Weise sagt und das der Intersprechervariabilität, nämlich die Unterschiede zwischen verschiedenen Sprechern und die daraus resultierenden unterschiedlichen Aussprachen derselben Wörter. Einige der Differenzen, wie beispielsweise die unterschiedliche Stimmlage (beschrieben durch die sogenannte Grundfrequenz) lassen sich relativ gut kompensieren. Andere, wie die Unterschiede zwischen Muttersprachlern und fremdsprachig Aufgewachsenen, müssen durch entsprechende Trainingsdatensätze neu gelernt oder das Modell im Nachhinein auf die einzelnen Sprecher adaptiert werden.

Weitere Schwierigkeiten ergeben sich direkt aus den Eigenschaften von Sprachen. Zu dieser Gruppe zählen beispielsweise Homophone. Es ist nicht realisierbar, anhand des Sprachsignals das richtige Wort zu identifizieren, wenn mehrere gleichklingende Wörter vorliegen. Dazu braucht es ein Sprachmodell: im Sprachmodell muss das richtige Wort für den Kontext gefunden werden. Eine weitere Herausforderung betrifft die orthographische Ebene. Die Erkennung und richtige Setzung von Wortzeichen beeinflusst auch die Resultate der Spracherkennung.

2.1.6 Evaluation eines Spracherkennungssystems

Die Qualität der Spracherkennung kann mit verschiedenen Messgrössen evaluiert werden. Eine der häufigsten Messgrössen ist die sogenannte Word Error Rate (WER). Die Berechnung von WER basiert auf der Anzahl Einfügungen (Insertions), Auslassungen (Deletions) und Ersetzungen (Substitutions). Wenn ein Wort in der maschinellen Transkription existiert, welches in der manuellen Transkription nicht vorhanden ist, handelt es sich um eine Insertion. Ein Fall, in dem ein Wort aus der manuellen Transkription in der maschinellen Transkription fehlt, wird als Deletion betrachtet. Wenn hingegen in der maschinellen Transkription am gleichen Ort ein anderes Wort steht, als in der menschlichen Transkription, wird es als Substitution betrachtet. WER wird wie folgt berechnet:

$$WER = \frac{S + D + I}{N}$$

Wobei S für Anzahl Substitutions, D für Anzahl Deletions, I für Anzahl Insertions und N für Anzahl Wörter in der manuellen Transkription steht. Eine weitere Messgrösse ist die Word Recognition Rate (WR). Bei der Berechnung von WR werden die Insertions nicht als Fehler betrachtet. Die Formel für WR lautet:

$$WR = \frac{C}{N} * 100\%$$

Wobei C für korrekt erkannte Wörter in der maschinellen Transkription und N für die Anzahl Wörter in der manuellen Transkription steht.

2.1.7 Aktueller Stand der Technik

Die grössten Technologie-Konzerne wie Google, IBM und Microsoft optimieren kontinuierlich ihre Speech Recognition Technologien, um den WER-Wert zu minimieren und eine möglichst gute Qualität der Spracherkennung zu erzielen. Im Jahr 2017 haben alle drei Unternehmen bekanntgegeben, dass sie die Fehlerquoten von ihren Speech-Recognition-Systemen im Vergleich

zu vergangenen Jahren senken konnten. In März erklärte IBM, dass die von ihnen erreichte WER neu 5.5% beträgt. 2 Monate später hat Google mitgeteilt, dass sie die WER auf 4.9% reduzieren konnten. Und in August gab Microsoft ihre neue Fehlerquote von 5.1% bekannt. Es wurde allerdings nicht von allen Konzernen erwähnt, auf welchen Korpusse die WER erreicht wurde. In der Regel wird der "Switchboard" Korpus dazu genutzt, Benchmarks für die Spracherkennungssysteme zu bestimmen. Bei Switchboard handelt sich um eine Kollektion von aufgenommenen Telefongesprächen zu alltäglichen Themen. IBM hat die oben genannte WER auf dem Switchboard erreicht [5]. Ob es auch bei Google und Microsoft der Fall war, ist unklar. Die WER von 4.9% von Google ist momentan als das beste bisher erreichte Resultat eines Spracherkennungssystems zu betrachten und ist damit erstmals besser als die durchschnittliche Fehlerrate von manuellen Transkriptionen, welche bei 5.0% liegt.

2.2 Sprechertrennung

Sprechertrennung ist ein Überbegriff für drei separate und unterschiedlich komplexe Problemstellungen. Sprecher-Verifikation ist die erste und einfachste Einstellung. Dabei handelt es sich um eine Boole'sche Funktion, bei der entschieden wird, ob eine Aussage zu einem gegeben Sprecher-Modell gehört, oder nicht. Die zweite Problemstellung ist die Sprecher-Identifikation oder Sprecher-Erkennung. Es wird eine Aussage mit verschiedenen Sprecher-Modellen verglichen, mit dem Ziel, den korrekten Sprecher ausfindig zu machen. Das letzte und komplexeste Verfahren ist die Sprecher-Diarization, was die beiden anderen beinhaltet. Bei diesem Fall sind keine Sprecher-Modelle im Vorhinein vorhanden, was bedeutet, dass anhand der Merkmale der Audio-Datei allein verschiedene Sprecher unterschieden werden müssen.

2.2.1 Sprecher-Segmentierung

Sprecher-Segmentierung ist der Prozess der Identifikation von Sprecherwechsel in einem Audio-Stream und der Aufteilung des Streams in akustisch gleichwertige Segmente, welche idealerweise nur einen Sprecher beinhalten. Dieser Prozess ist der erste von zwei Schritten in der Sprecher-Diarization.

Um einen Audio-Stream zu segmentieren, ist es notwendig, sogenannte Features aus dem Audio-Signal zu extrahieren, welche das Signal charakterisieren (vgl. Abschnitt 2.1.2). Dabei werden die relevanten Informationen beibehalten oder sogar verstärkt, während die irrelevanten Informationen für Segmentierung wegfallen [7]. Die Feature Extraktion wird nicht direkt auf dem Audio-Signal durchgeführt, sondern das Signal wird in überlappende Frames von je einigen Millisekunden aufgeteilt. In der Sprach- und Sprecher-Erkennung verbreitete Features sind unter anderem: Lautstärke, Tonlage, Zero Crossing Rate und Spektrale Features wie die Mel-Frequenz-Cepstrum-Koeffizienten (MFCCs) [15].

Anhand der erarbeiteten Features kann das Audio Signal segmentiert werden. Ein Ansatz dazu wäre eine energiebasierte Entscheidung, d.h. wie sich die Lautstärke von Frame zu Frame verhält, jedoch hat sich in einem Vergleich von Kemp et al. gezeigt, dass modell- und metrikbasierte Entscheidungen wesentlich bessere Resultate liefern [16]. Modellbasierte Algorithmen setzen voraus, dass im Vorhinein Modelle für verschiedene akustische Klassen trainiert werden, mit welchen der Audio-Stream in sprachliche und nicht sprachliche Segmente aufgeteilt wird [8], während metrikbasierte Algorithmen keinerlei Vorbereitung brauchen. Diese Algorithmen

«schieben» zwei Analyse-Fenster über den Audio Stream und vergleichen deren Inhalte mittels einer Distanz Funktion. Wenn das lokale Maximum über einem Schwellenwert liegt, dann wird segmentiert [9,14].

2.2.2 Sprecher Clustering

Nachdem ein Audio-Stream segmentiert wurde, müssen die einzelnen Segmente entsprechend ihrer Merkmale gruppiert werden. Dadurch lässt sich die Anzahl Sprecher sowie die Zugehörigkeiten der Äusserungen bestimmten.

Die erste Kategorie von Clustering-Methoden sind die deterministischen. Die Idee dahinter ist, die Audio-Segmente in denen gesprochen wird, gemäss ihrer Ähnlichkeit in einer Metrik zu clustern.

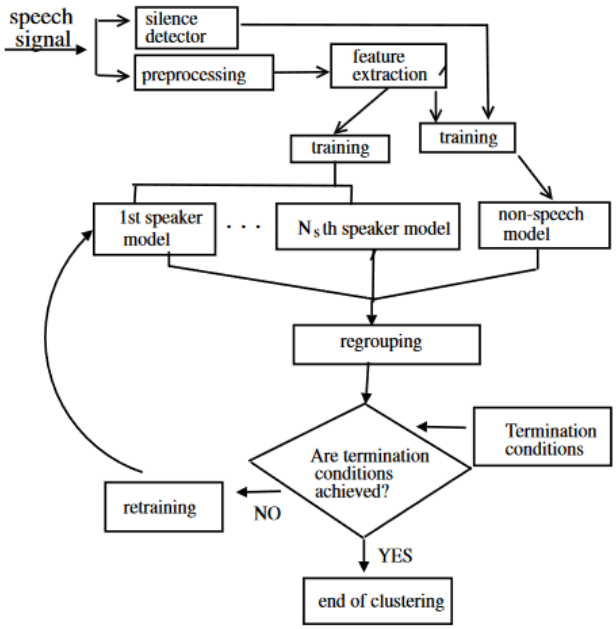


Abbildung 3: Allgemeiner Ablauf eines unüberwachten Sprecher Gruppierung-Systems [14]

SOM (self-organizing map), eine Art künstliches neuronales Netzwerk, das mit unüberwachtem Lernen trainiert wird, hat sich für deterministisches Sprecher-Clustering bewährt [10]. Die Funktionsweise ist grundsätzlich in Abbildung 3 wiedergeben. Mit dieser Methode sind keine Segmentierungsalgorithmen notwendig, sondern der Audio-Stream wird in kurze Segmente zerschnitten, welche dann geclustert werden. Ebenfalls zu bemerken ist, dass die Anzahl Sprecher für diesen Prozess bekannt sein muss [14]. Die zweite Kategorie ist die der probabilistischen Clustering-Methoden, welche Gaussian Mixture Models (GMM) oder Hidden Markov Models (HMM) verwenden (vgl. Abschnitt 2.1).

2.2.3 Aktueller Stand der Technik

Sprecher Diarization wird zurzeit bereits von mehreren Sprache-zu-Text-Diensten mitangeboten. IBM mit Watson, wie auch Google mit Google Cloud bieten beide Diarization als Option in ihren Services mit an. Diarization bei beiden Anbietern befindet sich jedoch noch im Beta Stadium und unterstützt werden erst Englisch und Spanisch sowie Japanisch auf IBM. Während IBM noch keine Informationen zu ihrer Diarization Error Rate (DER) [11] publiziert hat, hat Google in einem

kürzlich veröffentlichten Paper [12] bekanntgegeben, dass sie mit überwachtem maschinellem Lernen die DER des vorherigen clusterbasierten Ansatzes um 1.2% auf 7.6% reduzieren konnten.

3 Evaluation bestehender Tools

Im Folgenden wird zuerst ein Überblick über die verfügbaren, kompletten Transkriptions-Lösungen für den Endbenutzer geschaffen. Darauf werden Speech-to-Text Dienste, wie auch Diarization Dienste genauer analysiert, um den optimalen Service für einen ersten Prototypen bestimmen zu können.

3.1 Transkriptions-Tools für Endbenutzer

Untersucht wurden fertige Software Lösungen, die entweder online oder offline von einem Endbenutzer zur Transkription einer Audio-Aufnahme verwendet werden kann.

3.1.1 Spracherkennung mit Sprecher Diarization

Etablierte und einfache Software Lösungen die Audio-Dateien transkribieren können und gleichzeitig Sprecher Diarization unterstützen gibt es zurzeit nicht. Dienste wie die «Dragon» Suite von Nuance [13] lassen die Diarization weg und fokussieren sich auf Transkription in Echtzeit oder von fertigen Audio-Dateien. Die meisten Enterprise Lösungen verwenden eine Mischung von automatisierter Transkription mit externen Diensten und menschlicher Transkription für erhöhte Präzisionsanforderungen.

3.1.2 Fazit

Obwohl die Transkriptionsqualität bei allen Diensten hoch ist, wird bei allen die Sprechererkennung ignoriert. Preislich variieren die Angebote stark, von einmaligen Zahlungen für eine Software Suite [13], zu Preisen pro Minute Audio mit verschiedenen Qualitätsstufen [14].

3.2 Toolkits und Lösungen für Entwickler

Für die Implementation des Interview-Transkription Prototypen ist Funktionalität zur Spracherkennung und zur Sprecher Diarization nötig. In diesem Kapitel werden mehrere bereits bestehende Dienste und Tools verglichen, um den bestmöglichen Ansatz für das Programm zu finden.

3.2.1 Spracherkennung

Bei der Spracherkennung wurden drei Sprach-zu-Text Dienste in Betracht gezogen. Im Folgenden werden drei Dienste, Google Cloud STT, IBM Watson STT und Mozilla DeepSpeech in acht Kategorien verglichen, um den optimalen Service für einen Prototypen ausfindig zu machen. Die untersuchten Kriterien sind die unterstützten Sprachen, ob «Confidence» Werte vorhanden sind, die Qualität der Spracherkennung (Word-Error-Rate), ob die Sprache automatisch erkannt wird, ob Diarization möglich ist, welche Input-Formate möglich sind, welche Limitation beim Input gegeben sind, wie der Output aussieht, wie die Kosten gestaltet sind und ob sich das Tool auch offline verwenden lässt.

	Google Cloud Speech-to-Text	**IBM Watson Speech-to-Text**	**Mozilla DeepSpeech**
Unterstützte Sprachen	120 Sprachen inklusive Deutsch	Brasilianisches Portugiesisch,	Von Mozilla wird nur ein Englisches

	und verschiedene Varianten von Englisch	Französisch, Deutsch, Japanisch, Koreanisch, Mandarin (Chinesisch), Modern Standard Arabisch, Spanisch, Englisch (UK & US)	Sprach-Modell zur Verfügung gestellt, aber Dritte haben weitere Modelle für weitere Sprachen publiziert
«Confidence» pro Wort vorhanden	Ja	Ja	Nein
Qualität STT	Aktuellste Word-Error-Rate: 4.9%	Aktuellste Word-Error-Rate: 5.5%	Keine offiziellen Angaben, aber eigene Versuche haben mit dem Englischen Sprach-Modell von Mozilla schlechte Resultate ergeben
Automatische Erkennung der Sprache	Ja, aber manuelle Angabe ist ebenfalls möglich	Nein	Nein
Sprecher Diarization (und Anzahl Sprecher falls ja)	Im Beta Zustand für Englisch und Spanisch ohne Angaben zur maximalen Anzahl Sprecher	Im Beta Zustand für Englisch, Spanisch und Japanisch mit einer maximalen Anzahl Sprecher von sechs	-
Unterstützte Input-Formate	WAV Format, beliebige Abtastrate, synchrone und asynchrone Verarbeitung wird unterstützt	WAV, MP3, FLAC und mehr sind möglich,	WAV Format
Grösse der Input-Dateien	Mit asynchroner Verarbeitung bis zu 180 Minuten	Maximal 100 MB	Beliebig
Output Formate	JSON	JSON	Fliesstext
Kosten	1.43 CHF / Stunde Audio mit 0.020 – 0.035 CHF / GB und Monat für Speicher.	1.65 CHF / Stunde Audio	Keine (Lizenziert unter Mozilla Public License 2.0)
Online / Offline	Online	Online	Offline
Weitere Vorteile	Viele Optionen und mitgelieferte Meta-Informationen die alle gut Dokumentiert	Viele gut dokumen-tierte Meta-Informa-tionen sowie	Unterstützt eigene Sprach-Modelle und ist in Python geschrieben, was

	sind sowie dedizierte Sprach-Modelle für verschiedene Inputs (z.B. Telefon-gespräch)	Schlüsselwort-Erkennung	Modifikationen erlauben würde
Weitere Nachteile	Grössere Audio-Dateien werden automatisch in der Google Cloud gespeichert, was zu zusätzlichen Kosten führt	-	-

Tabelle 1: Vergleich von Spracherkennungsdiensten

3.2.2 Sprecher Diarization

Bei der Sprecher Diarization fallen die untersuchten Tools und Dienste in zwei Kategorien: Online und Offline Lösungen. In die erste Kategorie fallen die integrierten Diarization Dienste von Google, IBM und CMU Sphinx. Die primäre Funktion aller drei ist Speech-to-Text, jedoch lässt sich Sprecher Diarization als optionales Feature aktivieren. Wie in Abschnitt 1.1.3 bereits erklärt, ist die Funktionalität erst in wenigen Sprachen verfügbar und die Diarization-Error-Rate bewegt sich zwischen 5 und 10 Prozent.

Die zweite Gruppe sind Toolkits, mit denen sich Audio-Dateien offline segmentieren und diarisieren lassen. Folgende wurden für diese Arbeit in Betracht gezogen:

- ALIZE ist eine Sammlung mehrerer Open-Source-Komponenten, mit denen sich Sprecher Diarization durchführen lassen. Die Lösung wurde in C++ an der Universität in Avignon implementiert [15].
- AudioSeg ist ein Toolkit zur Segmentierung und Klassifikation von Audio-Streams. Es implementiert energiebasierte Stille-Erkennung, BIC Segmentierung und Clustering sowie GMM / HMM Klassifikation. Die Binärdateien wurden in C für Linux Systeme programmiert.
- Die Python Bibliothek pyAudioAnalysis ermöglichtet eine grosse Anzahl Audio-Analyse-Verfahren, darunter Stille-Erkennung und Sprecher Diarization.

3.2.3 Fazit

Der Prototyp soll mit dem Google Cloud Speech-to-Text Dienst implementiert werden. Die Infrastruktur der Unternehmung garantiert hohe Genauigkeit bei der Spracherkennung, und Aufgrund der direkt integrierten Sprecher Diarization fällt einiges an zusätzlichem Aufwand weg. Bei den folgenden Iterationen dieser Arbeit wo Unterstützung für Deutsch eine zentrale Rolle spielt, muss die Diarization mittels eines anderen Tools gelöst werden. Weil die Applikation schlussendlich völlig offline verfügbar sein soll, würde sich dort eine Kombination von Mozilla DeepSpeech für die Spracherkennung und einem Tool wie ALIZE für Diarization anbieten.

4 Vorgehen

Im Folgenden werden die Anforderungen und die primären User Stories, das Design anhand der physischen und logischen Architektur und zuletzt die implementierte Lösung erläutert.

4.1 Anforderungen

Das Kapitel Anforderungen teilt sich auf in funktionale und nicht-funktionale Anforderungen. Die funktionalen Anforderungen bestehen ausschliesslich aus User Stories.

4.1.1 Funktionale Anforderungen: User Stories

Folgende User Stories beziehen sich auf insgesamt vier Szenarios. Zum einen sollen Audio-Dateien transkribiert, strukturiert und angezeigt werden (US1 – US3). Zusätzlich soll das Audio angehört werden können und es soll gleichzeitig im Text ersichtlich sein, wo man sich befindet (US4, US5). Ausserdem sollen Transkripte exportierbar gemacht werden (US10, US11).

US1	**Als Interviewer möchte ich eine Interview Aufzeichnung hochladen und die Transkription angezeigt bekommen, damit ich dies nicht von Hand erledigen muss.**
Akzeptanz Kriterien	• Transkription: Das Transkript steht zur Verfügung nachdem die Datei hochgeladen und verarbeitet wurde • Aufnahmen bis zu 60 Minuten • In englischer Sprache • Nur .wav Format

US2	**Als Interviewer möchte ich das die Transkription in Sprecherabschnitte strukturiert ist, damit ich klar unterscheiden kann, welche Äusserungen zu welchen Sprechern gehören.**
Akzeptanz Kriterien	• Sprecheränderungen müssen für mindestens zwei Sprecher richtig erkannt werden. • Jeder Äusserung ist ein Sprecher-Id zugeordnet. • Die Äusserungen derselben Sprecher müssen die gleiche Id besitzen.

US3	**Als Interviewer möchte ich den Sprechern Namen geben, damit nicht nur eine Sprecher-Id angezeigt wird und ich die Sprecher besser auseinanderhalten kann.**
Akzeptanz Kriterien	• Wenn ein Sprecher ein Name besitzt, muss dieser im Transkript angezeigt werden, ansonsten wird die Sprecher-Id angezeigt.

US4	**Als Interviewer möchte ich das Audio abspielen können und die dazugehörigen Worte im Transkript sollen passend zum Audio hervorgehoben werden.**

Akzeptanz Kriterien	• Es muss erkennbar sein, welche Worte gerade gesprochen werden.

US5	**Als Interviewer möchte ich durch das Audio navigieren können indem ich die dazugehörenden Worte im Transkript anklicke, damit ich einfach falsch transkribierte Passagen finden und direkt anhören kann.**
Akzeptanz Kriterien	• Das Audio muss an der Stelle abgespielt werden, an der das dazugehörende Wort anfängt.

US10	**Als Interviewer möchte ich das Transkript exportieren und lokal speichern können, um ausserhalb der Anwendung weiter daran zu arbeiten.**
Akzeptanz Kriterien	• Das Export-Format (docx, txt, html, etc.) muss ausgewählt werden können.

US11	**Als Interviewer möchte ich das komplette Projekt speichern und wieder öffnen können, damit über längere Zeit am gleichen Transkript gearbeitet werden kann.**
Akzeptanz Kriterien	• Das Projekt muss gespeichert werden können. • Das Projekt muss geöffnet werden können. • Bei Änderungen am Projekt soll nach dem ersten Speichern automatisch gespeichert werden können.

Tabelle 2: Funktionale Anforderungen

4.1.2 Nicht-funktionale Anforderungen

Folgende nicht-funktionalen Anforderungen müssen erfüllt werden, damit die Software einfach und ohne grossen Aufwand eingesetzt werden kann:

NFR 1	**Ein Benutzer soll die Anwendung verwenden können, ohne komplizierte Installationen oder Einstellungen vorzunehmen.**
Akzeptanz Kriterien	• Es muss nur eine Installationsdatei ausgeführt werden, anschliessend kann die Anwendung gestartet und benutzt werden.

NFR 2	**Der Benutzer soll die Anwendung intuitiv bedienen können.**
Akzeptanz Kriterien	• Die Anwendung muss ohne Lesen einer Anleitung verständlich und bedienbar sein.

Tabelle 3: Nicht-funktionale Anforderungen

4.2 Design

Im Folgenden werden die physische und logische Architektur mit den eingesetzten Technologien, das Klassendiagramm und zuletzt die erarbeiteten Wireframes für die Benutzeroberfläche beschrieben.

4.2.1 Physische Architektur und eingesetzte Technologien

Das Ziel der gewählten Architektur ist die einfache Einsetzbarkeit der Software. Der Installationsaufwand soll minimiert werden und es sollen keine Einstellungen am System manuell vorgenommen werden müssen. Ausserdem soll die Architektur so gewählt werden, dass andere Speech-to-Text und Diarization APIs einfach und ohne grossen Programmieraufwand eingesetzt werden können. Damit soll auch die Möglichkeit gegeben werden, die Anwendung komplett offline zu verwenden, falls entsprechende APIs benutzt werden. Um diese Ziele zu erreichen wurde eine «All-in-One» Lösung verwendet. Alle Teilsysteme, abgesehen von der in diesem Prototypen verwendeten API, laufen direkt auf dem Client.

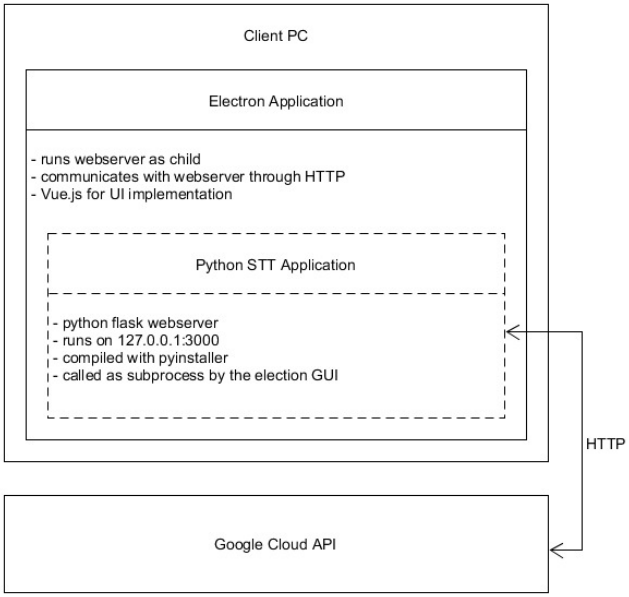


Abbildung 4: Architektur des Interview Transcription Tools

Das System besteht aus drei Teilen, nämlich der Python Applikation (Backend), der Electron Applikation (Frontend) und der verwendeten Transkriptions-API (in diesem Projekt wurde die Google Cloud API benutzt). Die Funktionsweise des Systems ist in Abbildung 4 dargestellt.

4.2.2 Logische Architektur und Klassendiagramm

Die Python Applikation ist die zentrale Schnittstelle zur Google Cloud API und ist zuständig, die Audio Datei an die Google Cloud API zu schicken, das verarbeitete Transkript zu parsen und mittels JSON an das Frontend zu senden. Python wurde ausgewählt, da diese Programmiersprache sehr einfach zu erlernen, gut einsetzbar und anerkannt für Machine Learning und Data Science Problemstellungen ist. Python ist mit den vielen Bibliotheken für jede Situation und Problem einsetzbar [17].

Um die Daten dem Frontend bereitzustellen, wird Flask [18] verwendet. Dies ist ein Framework für Python und wird als REST-Schnittstelle zwischen Python und Electron [19] für den einfachen Datenaustausch mittels Webserver über HTTP eingesetzt.

Electron wiederum ist die Umgebung, welche verwendet wird, um Desktop Anwendungen zu bauen. Electron verwendet Chromium und Node.js, um Webseiten darzustellen. Für die UI-Gestaltung wird das sehr moderne JavaScript Framework Vue.js [20] eingesetzt. Dies erlaubt, das Frontend als Single-Page Applikation zu gestalten und ist sehr einfach einzusetzen.

Damit umständliche Installationen und Einstellungen überflüssig werden, wird die Electron Anwendung gepackaged und zu einer ausführbaren .exe Datei umgewandelt. Dafür wird das Python Packaging Framework PyInstaller [21] verwendet.
Das Python-Backend ist folgendermassen aufgebaut (dem Klassendiagramm in Abbildung 5 zu entnehmen):

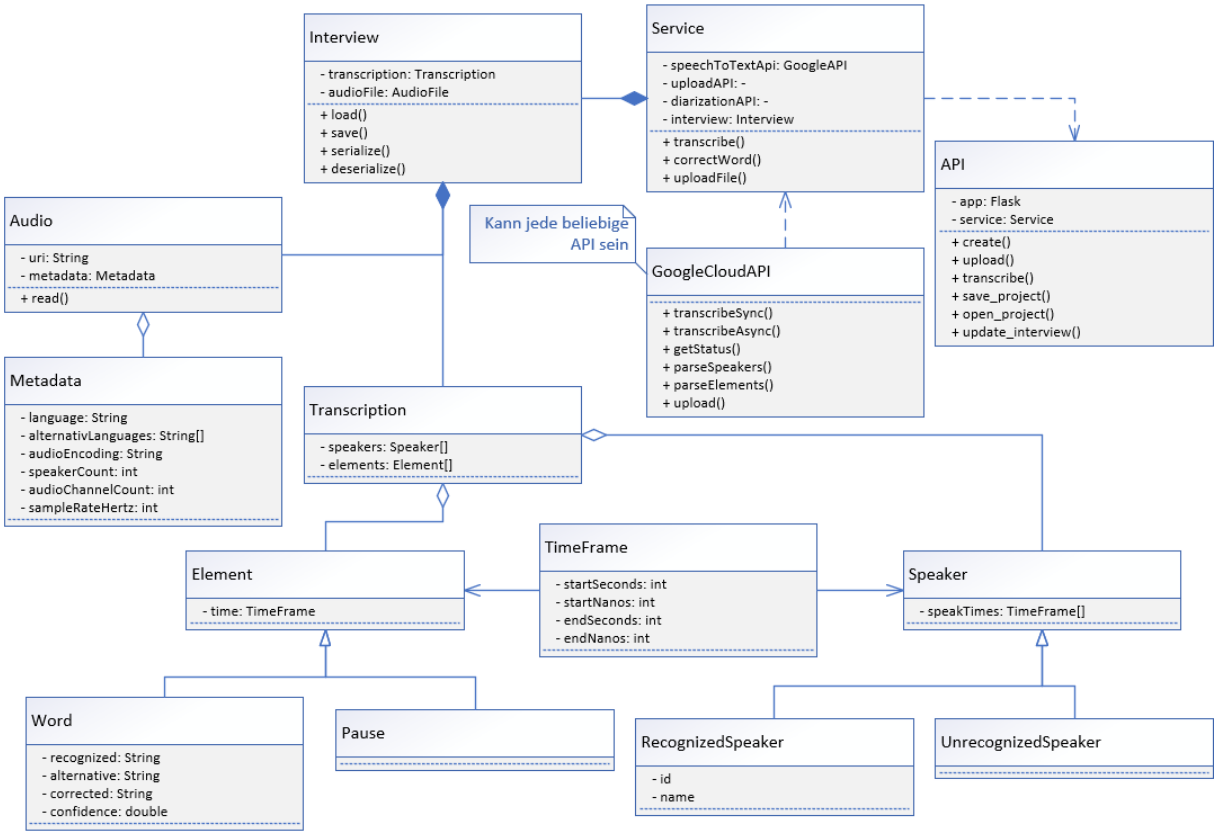


Abbildung 5: Klassendiagramm Python-Backend

Die Zentrale Schnittstelle zum Frontend ist die Klasse API. Diese Klasse beinhaltet die REST-Schnittstellen für das erstellen, hochladen und transkribieren der Interview Audio Datei. Zusätzlich sind die Methoden für das Speichern und Öffnen der Projekte und der Aktualisierung der Transkription in dieser Klasse zu finden. Sie beinhaltet ausserdem die Datenfelder für den Flask Server und den Service, welcher die Aufgaben an die darunterliegenden Klassen weiterleitet und das Interview mit der Transkription und dem Audio mit Metadaten (z.B.: Anzahl Sprecher, Audiofrequenz, Anzahl Audiokanäle, Sprache etc.) verwaltet. Die Transkription wiederum beinhaltet alle Sprecher und Elemente, wie Worte und Pausen. Jedes Element sowie jeder Sprecher besitzen Zeitangaben, in welchem Zeitraum sie im Transkript auftreten. Die Service Klasse

verwaltet auch alle erforderlichen APIs zum Hochladen der Audios, der Transkription der Interviews und der Sprecher Diarization. In dieser Arbeit ist dies lediglich die Google Cloud API.

4.2.3 Wireframes

Es wurden einzelne Skizzen (Wireframes) für die Oberfläche gezeichnet. Die Anwendung soll schlussendlich ungefähr so aussehen:

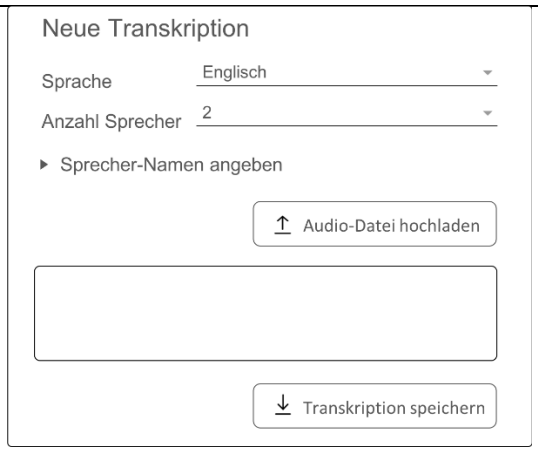
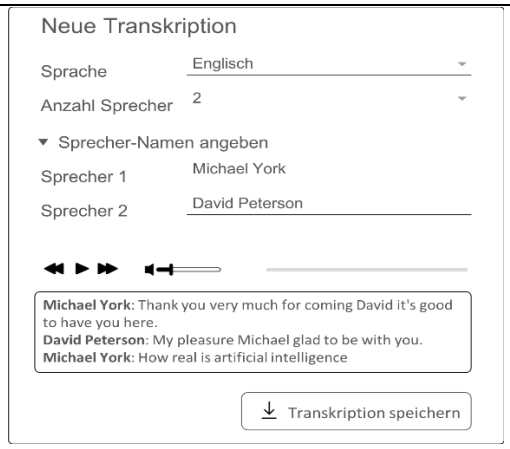
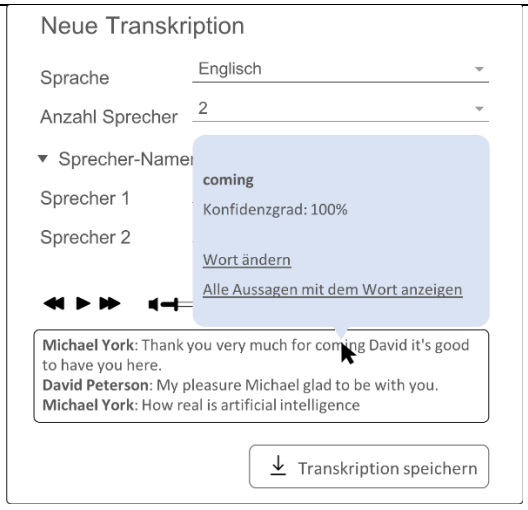
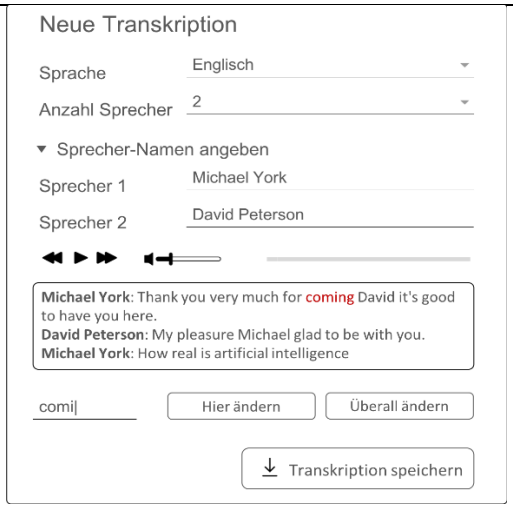
<table>
<tr>
<td> 
</td>
<td> 
</td>
</tr>
<tr>
<td>So sieht das User Interface aus, wenn noch kein Interview transkribiert wurde. Zur Transkription muss lediglich die Sprache, die Anzahl Sprecher und eine Audio Datei hochgeladen werden.</td>
<td>Nachdem eine Audio Datei hochgeladen und transkribiert wurde, werden die Sprecher, der Audio-Player und die Transkription angezeigt.</td>
</tr>
<tr>
<td> 
</td>
<td> 
</td>
</tr>
<tr>
<td>Durch einen Rechtsklick auf ein Wort kann das Kontextmenü geöffnet werden. Es wird die Wortkonfidenz angezeigt und die Möglichkeit, dieses Wort zu ändern oder alle gleichen Wortvorkommen im Text anzuzeigen.</td>
<td>Durch einen Klick auf «Wort ändern» kann es an einem Ort oder überall geändert werden.</td>
</tr>
</table>

Tabelle 4: Wireframes für das User Interface

4.3 Implementierte Lösung

Die Anwendung wurde über 14 Wochen hinweg implementiert mit der Beschränkung, dass nur amerikanisch-englisch gesprochene Audio Dateien transkribiert werden können. Der Grund dafür ist, dass die Google Cloud API in amerikanischem Englisch die meisten Funktionen unterstützt und am besten funktioniert.
Ausserdem ist die Oberfläche der Anwendung nur in Englisch verfügbar ist. Es wurden die User Story 1 bis User Story 5 und User Story 11 implementiert. Dafür wurde zuerst das Backend implementiert und mittels Testskripts getestet. Anschliessend wurde die Electron Umgebung integriert und das Frontend mit der Webseite aufgebaut.
Wie der Abbildung 6 entnommen werden kann, entspricht die Implementation nicht genau den Wireframes aus dem vorherigen Unterkapitel. Ein Grund dafür ist, dass sich diese Arbeit vor allem auf die Funktionalität und weniger auf das Aussehen der Anwendung konzentriert hat. Das User Interface wird in einer zukünftigen Arbeit angepasst und verbessert.

Folgender Ablauf bei einer Interview-Transkription kann der Abbildung 6 entnommen werden: Zuerst wird die zu transkribierende Audio Datei ausgewählt. Nach dem Auswählen wird die Audio-Datei vom Backend vorbereitet, das heisst es wird die Hertz-Frequenz und die Anzahl Audio-Kanäle analysiert und der Inhalt der Audio-Datei zurück an das Frontend gegeben, damit das Interview angehört werden kann.
Anschliessend muss die Sprache (hier nur «en-US» aus eben genannten Gründen möglich) und die Anzahl Sprecher gewählt werden. Danach kann das Audio transkribiert werden. Nach Abschluss des Vorganges wird die Transkription angezeigt, wobei die Sprecher-Abschnitte klar erkennbar sind. Wenn eine Pause erkannt wurde, wird dies mit eckigen Klammern und drei Punkten dazwischen angezeigt.
Unterhalb des Transkripts befindet sich ein Button, mit dem das Transkript in die Zwischenablage kopiert wird, damit der Text einfach weiterverwendet werden kann. Direkt unterhalb davon können die Namen der Sprecher geändert werden. Durch einen Klick auf den Button «Rename» aktualisiert sich das Transkript und zeigt die neuen Sprecher Namen entsprechend an. Schlussendlich kann das komplette Interview-Projekt gespeichert (Automatisches Speichern kann manuell gewählt werden) und ein bestehendes Projekt geöffnet werden. Durch Überfliegen der Maus von Worten oder Pausen wird ein Tooltip mit der Konfidenz oder der Pause in Sekunden angezeigt. Durch einen Klick auf ein Wort wird die entsprechende Stelle abgespielt und kann zur Überprüfung angehört werden.

Eine Installationsanleitung kann im Anhang unter Abschnitt 8.3.3 gefunden werden. Um die Anwendung zu testen sind auch Testdateien vorhanden, diese befinden sich auf dem beigelegten USB-Stick.

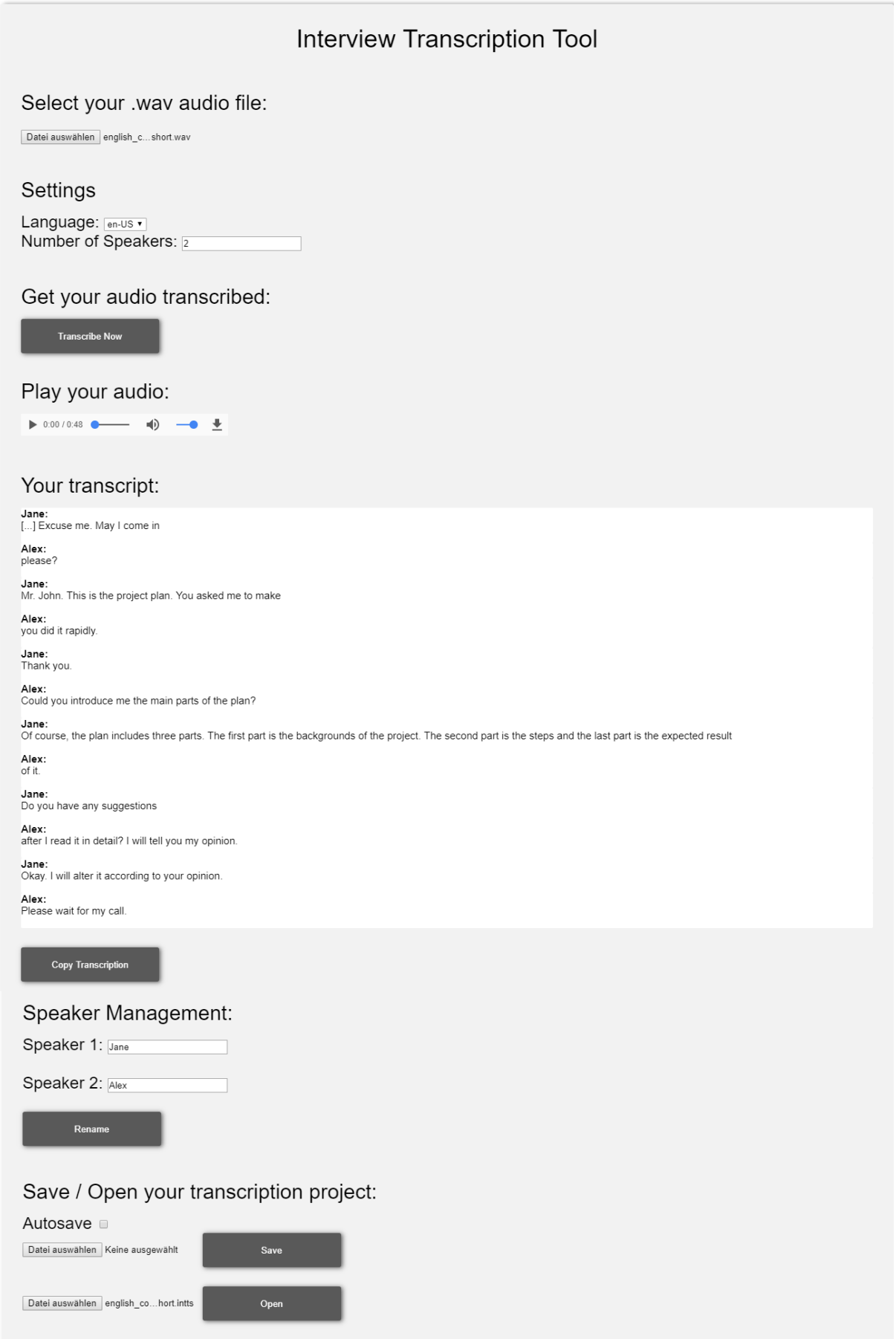


Abbildung 6: Screenshot der Implementation

5 Evaluation

Sobald die Implementierung der Grundfunktionen der Transkription Anwendung abgeschlossen wurde, wurden die Funktionen für die Spracherkennung und Sprecher-Diarization einer Evaluation unterzogen. Das Ziel der Evaluation war es, die Leistung des Interview Transkription Tools zu beurteilen, Fehler und Verbesserungsmöglichkeiten zu identifizieren und letztendlich die Ergebnisse mit den State-of-the-art-Lösungen zu vergleichen.

5.1 Evaluationsablauf

5.1.1 Theoretischer Hintergrund

Die im Projekt angewandte Evaluationsmethodik setzt sich aus 5 Schritten zusammen (vgl. Abbildung 7). Der erste Schritt besteht in der Bestimmung von Evaluations-Szenarien. Es wird entschieden, welche Parameter berücksichtigt werden müssen, damit die Evaluation möglichst viele Szenarien abdeckt. Anschliessend werden Messgrössen bestimmt, die es ermöglichen, die Leistung des Transkriptionstools zu messen und mit anderen bestehenden Tools zu vergleichen. Ferner wird nach einer geeigneten Evaluations-Software gesucht. Je nach der ausgewählten Software, wird im nächsten Schritt der Korpus entsprechend vorbereitet. Die im Korpus enthaltenen Dateien müssen entsprechend formatiert werden, um vom Evaluationstool verarbeitet werden zu können. Der letzte Schritt besteht in der eigentlichen Durchführung der Evaluation [1].

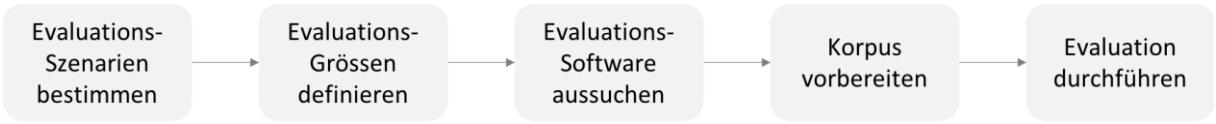


Abbildung 7: Evaluationsschritte nach [1]

Wie die Methodik in dem Projekt eingesetzt wurde und wie die einzelnen Schritte abgelaufen sind, wird in den folgenden Kapiteln beschrieben.

5.1.2 Evaluations-Szenarien bestimmen

Eins der Hauptziele bei der Bestimmung von Evaluations-Szenarien war es, die Qualität der Transkription im Zusammenhang mit der Qualität der Audio-Datei zu untersuchen. Daher wurden Merkmale wie Abtastrate, Anzahl Kanäle und Störungen bei der Szenarien-Bestimmung mit-einbezogen. Weitere Aspekte, die im Evaluationsprozess mitberücksichtigt wurden, waren Anzahl Sprecher und Sprechergeschlecht. Aus einer Zusammenstellung der erwähnten Merkmale ergaben sich 42 Evaluations-Szenarien:

- 12 Szenarien für 1 Sprecher, mit weiblichen und männlichen Sprechern, ohne Störungen, mit Abtastraten von 8000, 16000 oder über 40000 Hz, Mono und Stereo
- 24 Szenarien für 2 Sprecher, mit weiblichen und männlichen Sprechern gemischt sowie nur mit männlichen Sprechern, ohne und mit Störungen, mit Abtastraten von 8000, 16000 oder über 40000 Hz, Mono und Stereo
- 6 Szenarien für 3 Sprecher, mit weiblichen und männlichen Sprechern gemischt, ohne Störungen, mit Abtastraten von 8000, 16000 oder über 40000 Hz, Mono und Stereo

Eine detaillierte Liste von Evaluations-Szenarien befindet sich im Anhang.

5.1.3 Evaluations-Grössen definieren

Bei der Evaluation wurde die manuelle Transkription der Audio-Datei (der sogenannte Referenztext) mit der maschinellen Transkription (dem sogenannten Hypothesentext) verglichen. Bei dem Vergleich wurden 3 Messgrössen berechnet: Word Error Rate (WER) [1], Speaker Number Error Rate (SNER) und Utterance Number Error Rate (UNER). Die letzten zwei wurden für die Zwecke des Projektes definiert, damit eine Grundevaluation von Speaker-Diarization durchgeführt werden kann. In der Literatur wird in den meisten Fällen die DER-Messgrösse angewandt, wie im Abschnitt 2.2.3 erwähnt. Für die DER-Berechnung sind allerdings manuelle Transkriptionen mit genauen Zeit- und Dauerangaben der einzelnen Utterances erforderlich. Da der Korpus solche Transkriptionen nicht beinhaltet, musste auf die DER-Berechnung verzichtet werden.

Word Error Rate

Die Word Error Rate wird gemäss der im Abschnitt 2.1. beschriebenen Formel berechnet.

Speaker Number Error Rate (SNER)

Bei der Berechnung von Speaker Number Error Rate (Sprecheranzahl-Fehlerrate) wird die Anzahl Sprecher im Referenztext und die Anzahl Sprecher im Hypothesentext berücksichtigt:

$$SNER = 1 - \frac{SH}{SR}$$

wobei SH für Sprecheranzahl im Hypothesentext und SR für Sprecheranzahl im Referenztext steht.

Utterance Number Error Rate (UNER)

Als Utterance wird eine einzelne Äusserung eines Sprechers bezeichnet. Eine Utterance endet mit einem Sprecherwechsel. Nach einem Sprecherwechsel beginnt eine neue Utterance. Es wird der Unterschied zwischen der Anzahl Utterances in der Referenzdatei und in der Hypothesen-Datei für jeden Sprecher bestimmt. Die Utterance Error Rate (Äusserungs-Fehlerrate) wird wie folgt berechnet:

$$UER = \frac{\sum_{k=1}^{n} | UR_{S_k} - UH_{S_k} |}{UR}$$

wobei UH für Utterance-Anzahl im Hypothesentext und UR für Utterance-Anzahl im Referenztext steht. S_k bezeichnet den k-ten Sprecher. Mit der Berechnung der drei Messgrössen können die für die Evaluation festgelegten Ziele abgedeckt werden. Die Berechnung von WER erlaubt es, die Qualität der Spracherkennung zu beurteilen und die Messgrössen SNER und UNER ermöglichen eine Grundevaluation von Sprecher Diarization.

5.1.4 Evaluations-Software aussuchen

Es wurde nach einem bestehenden Open-Source-Evaluationstool gesucht, dass mindestens eine Speech-to-Text-Qualitätsevaluation anbietet. Es wurden einige Lösungen kurz untersucht, unter anderem Open-Source-Evaluationstools in Python und ein NIST SCTK Scoring Tookit. Nach einer kurzen Untersuchung wurde das Tool sclite [2] aus dem NIST SCTK Scoring Tookit ausgewählt. Es ist eine Open-Source-Software, die die Berechnung von WER durchführt,

Statistiken zu Referenztext-Hypothesen-Textpaare generiert und alle Ersetzungen, Löschungen und Einfügungen in einem Bericht auflistet. Es ist ein fertiges, relativ gut dokumentiertes Tool, das in mehreren wissenschaftlichen Arbeiten [10, 12, 13, 14, 15] als Tool zur WER-Berechnung eingesetzt wurde. Aus den Gründen wurde es für die Evaluation des Interview Transkription Tools ausgewählt.

Eingesetzte Lösung: sclite

Das sclite Tool vergleicht die manuelle Transkription mit der maschinellen, von einer Spracherkennungs-Software durchgeführten Transkription. Für den Vergleich sind 2 Dateien notwendig: eine Referenzdatei mit der manuellen Transkription (eine .ref-Datei) und eine Hypothesendatei mit der maschinellen Transkription (eine .hyp-Datei). Der Vergleich der .ref- und .hyp-Dateien wird als Alignment bezeichnet. Das Resultat des Alignments sind Statistiken und Berichte, die die Leistung der Transkriptionsanwendung auswerten und zusammenfassen [2].
Die Transkriptionen, die an das Tool als Inputs übergeben werden, müssen ein entsprechendes Format haben. Es stehen mehrere Formate zur Verfügung. Für die hier beschriebene Evaluation wurde das TRN Format ausgewählt. Im TRN-Format werden die Utterances durch Zeilenumbrüche separiert. Am Ende jeder Utterance steht eine Sprecher-ID zusammen mit der jeweiligen Utterance-ID.

Als Output werden Statistiken und Berichte für jedes Referenz-Hypothese-Paar generiert. Bei der Evaluation der Transkriptionsanwendung wurden jeweils 2 Dateien generiert. Zum einen eine .sys-Datei mit Statistiken zu WER, Insertions, Deletions und Substitutions. Zum anderen ein Bericht in Form einer .sgml-Datei, in der alle Wörter aus dem Referenztext mit ihren Entsprechungen aus dem Hypothesentext sowie alle Deletions und Insertions aufgelistet wurden. Der Bericht ermöglichte eine manuelle Analyse der sogenannten Confusion Pairs (Wörter, bei deren Transkription ein Fehler aufgetreten ist).

5.1.5 Korpus vorbereiten

Damit eine Evaluation nach den vordefinierten Szenarios mit der ausgewählten Software durchgeführt werden konnte, musste ein entsprechender Korpus zusammengestellt werden. Es wurde nach Audio-Dateien mit Transkriptionen gesucht, welche die im Kapitel 5.1.1 beschriebenen Kriterien erfüllen. Als Quellen wurden Audio-Bücher und E-Books sowie Podcasts von IEEE mit Transkriptionen verwendet.

Der Speech-Korpus entstand aus 5 Audiobücher und 13 Interviewsaufnahmen. Da nur .wav-Dateien als Input übergeben werden konnten, musste jede Audio-Datei im Korpus entsprechend konvertiert werden. Es wurden dabei mehrere Varianten der Audio-Dateien generiert: Jedes Audiobuch wurde in insgesamt 6 Varianten mit Abtastraten von 8000, 16000 und über 40000, jeweils Mono und Stereo, erstellt. So entstanden 30 Audio-Dateien mit einem Sprecher, entweder einem weiblichen oder einem männlichen. Jedes Interview wurde ebenfalls in 6 Varianten mit Abtastraten von 8000, 16000 und über 40000, jeweils Mono und Stereo, konvertiert. Als Resultat wurden 78 Audio-Dateien mit 2 bis 3 weiblichen oder männlichen Sprechern erstellt. Zu der Kollektion gehörten auch Interviews, die telefonisch durchgeführt wurden, und somit als Audioaufnahmen mit Störungen klassifiziert wurden.

Alle Dateien im Korpus werden nach dem gleichen Schema benannt:
[Anzahl Sprecher]_[Titel]_[Abtastrate]_[Anzahl Kanäle], z.B. "1_emma_16000_mono". Eine komplette Liste der Titel mit Angaben der Autoren befindet sich im Anhang. Insgesamt wurden 108 .wav-Dateien für die Evaluation zur Verfügung gestellt, mit der Dauer von insgesamt fast 18 Stunden.

5.1.6 Evaluation durchführen

Die Referenzdateien wurden aus den Originaltranskriptionen durch Konvertierung in das erforderliche TRN-Format erstellt. Es wurden dabei auch mehrere Preprocessing-Operationen durchgeführt, wie Entfernung von Interpunktionszeichen oder überflüssigen Zeilenumbrüchen. Die .hyp-Dateien wurden aus den transkribierten Texten, welche von der Transkriptions-anwendung berechnet wurden, erzeugt. Diese mussten auch den gleichen Preprocessing-Operationen wie die .ref-Dateien unterzogen werden. Sie wurden auch in das TRN-Format konvertiert. So entstanden die .hyp-.ref-Paare, die anschliessend an das sclite Tool übergeben wurden.

Das sclite Tool führte das Alignment von .ref-Datei und .hyp-Datei durch und erzeugte als Resultat für jedes .ref-.hyp-Paar 2 Dateien: .sys und .sgml. Die Dateien wurden dann ausserhalb von sclite weiterverarbeitet. Aus den WER-Statistiken zu einzelnen Paaren wurden allgemeine Statistiken zu dem ganzen Korpus erstellt. Daneben wurden auch Statistiken zu der Diarization-Qualität generiert.

Im letzten Schritt wurde eine manuelle Analyse von ausgewählten .sgml-Dateien sowie .ref-.hyp-Paaren durchgeführt. Sie hatte zum Ziel, ein besseres Verständnis von Fehlerursachen zu gewinnen und die Herausforderungen der Spracherkennung in der Praxis zu beobachten.
Die Korpus-Erstellung sowie der Evaluations-Prozess sind unten grafisch dargestellt. Die blau markierten Kästchen stellen manuelle Schritte dar. Die grün markierten Schritte wurden automatisiert und können für einen beliebigen Korpus durchgeführt werden.

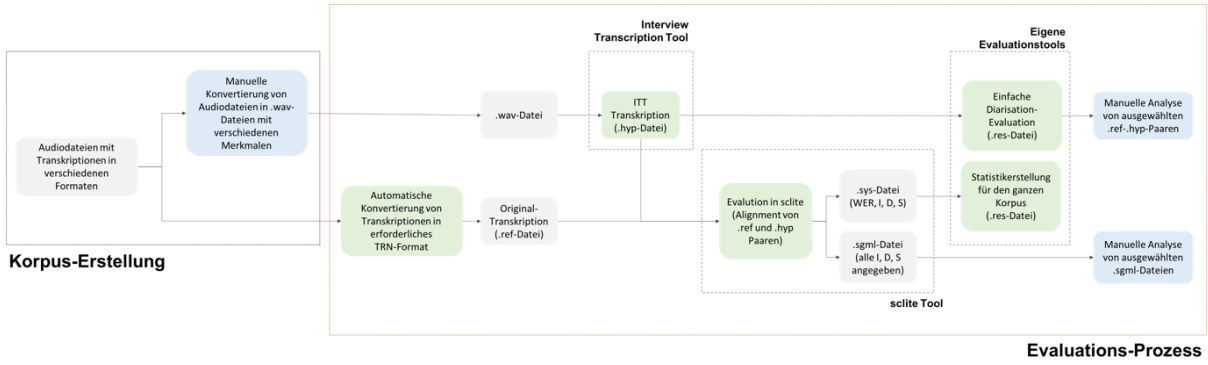


Abbildung 8: End-2-End Evaluations-Prozess der Interview Transkriptions Applikation

Die Resultate der automatischen Evaluation und der manuellen Analyse werden im nächsten Kapitel besprochen.

5.2 Evaluationsergebnisse und Schlussfolgerungen

Die Evaluation wurde separat für die Spracherkennung und für die Diarization durchgeführt. Den grössten Teil der Evaluation bilden automatisch generierte Statistiken zu den Messgrössen, die im Kapitel 2.1.2 beschrieben wurden. Darüber hinaus wurden auch manuelle Analysen von ausgewählten Transkriptionen und sclite-Berichten durchgeführt.
Bei der Evaluation wurde Bezug auf die im Kapitel 1.1.4 erwähnten Herausforderungen aus dem Bereich Spracherkennung genommen. Es wurde versucht, die Zusammenhänge zwischen den Ergebnissen und den besonders anspruchsvollen Aspekten der Spracherkennung zu erkennen.

5.2.1 Automatische Spracherkennung-Evaluation

Alle Transkriptionen wurden zuerst einer Evaluation hinsichtlich der Qualität der Spracherkennung unterzogen. Bei Interviews wurde dabei die Sprechertrennung ignoriert.
Den für den gesamten Korpus erstellten Statistiken kann man entnehmen, dass der kleinste erreichte WER-Wert 6.9% beträgt. Es handelt sich dabei um eine Transkription von einem Audiobuch, das von einem weiblichen Sprecher vorgelesen wird. In der Audioaufnahme treten keine Störungen vor, die Abtastrate beträgt 16000Hz und es ist ein Mono-Audio. Das schlechteste Resultat hingegen ist eine Fehlerrate von 225.1%. Der grosse WER-Wert ist allerdings auf einen Fehler in dem Transkriptionsprozess zurückzuführen. Bei der Transkription werden einige Textsegmente dupliziert, was die Anzahl Insertions extrem steigert. In dem oben erwähnten Fall beträgt die Anzahl Insertions 214.4%.

Es wurde beobachtet, dass der WER-Wert mehr vom Sprecher beziehungsweise von dem Inhalt der Aufnahme abhängt, als von den Merkmalen der .wav-Datei (Abtastrate und Anzahl Kanäle). Eine Liste von allen Transkriptionen sortiert nach dem WER-Wert weist folgende Eigenschaft auf: Die Transkriptionen bilden Cluster nach inhaltlichen Gemeinsamkeiten und nicht nach den Audiomerkmalen. Zur Veranschaulichung wird unten eine Liste der 25 besten WER-Resultate angeführt.

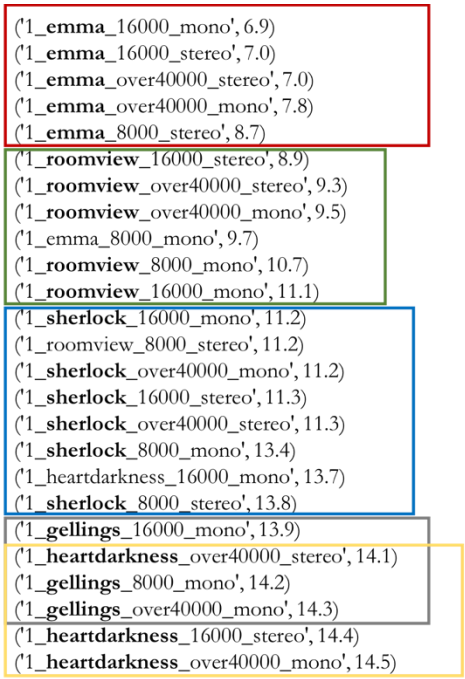


Abbildung 9: Die besten 25 Transkriptionsresultate nach dem WER-Wert sortiert

Aus der Liste kann man auch ablesen, dass die Transkriptionen von Audiobüchern, die nur einen Sprecher enthalten, bessere Genauigkeit der Transkription aufweisen, als Interviews. Das erste Interview (1_gellings_16000_mono) erscheint erst an der Stelle 20 mit einer WER von 13.9. Es ist ein Interview zwischen einer Sprecherin und einem Sprecher, wobei interessanterweise der Sprecher an dem Gespräch über eine Telefonverbindung teilnimmt, was die Qualität der Transkription beeinträchtigen soll. Die Qualität ist allerdings besser als bei anderen Interviews, die nicht telefonisch durchgeführt wurden. Der mögliche Grund ist die Sprechgeschwindigkeit. Es ist allerdings nur eine Vermutung, die auf der subjektiven Wahrnehmung des menschlichen Gehörs basiert.

In der kompletten Liste der WER-Werte wurde noch ein Ausreisser beobachtet: an der Stelle 39 kommt noch ein Audiobuch mit einem Sprecher (1_twocities_16000_mono), das einen WER-Wert von 21.8% hat. Das schlechte Resultat kann wieder an der Sprechgeschwindigkeit liegen oder eventuell am Wortschatz, das subjektiv betrachtet mehr anspruchsvoll ist, als bei anderen Audiobüchern. Es wurde nicht weiter untersucht.

Weitere Beobachtungen beziehen sich auf den Zusammenhang zwischen der Anzahl Kanäle und dem WER-Wert. Bei 11 von 18 Titeln war die Transkription am besten, als die zu transkribierenden .wav-Dateien in einer Stereo-Version vorlagen. Die durchschnittliche WER bei Stereo liegt allerdings höher als bei Mono: 42.18% versus 26.51%. Aus der vorliegenden Evaluation kann man also eine Schlussfolgerung ziehen, dass kein klarer Zusammenhang zwischen der Anzahl Kanäle und der Qualität der Transkription besteht. Die Tabelle 5 soll dies veranschaulichen. Sie enthält 6 Listen zu jeweils einem Titel, die nach dem WER-Wert steigend sortiert sind:

1_emma	1_twocities	1_brownell
1_emma_16000_**mono**, 6.9	1_twocities_16000_**mono**, 21.8	1_brownell_over40000_**mono**, 23.1
1_emma_16000_**stereo**, 7.0	1_twocities_16000_**stereo**, 22.0	1_brownell_over40000_**stereo**, 23.2
1_emma_over40000_**stereo**, 7.0	1_twocities_over40000_**stereo**, 22.2	1_brownell_16000_**mono**, 23.2
1_emma_over40000_**mono**, 7.8	1_twocities_over40000_**mono**, 22.7	1_brownell_8000_**mono**, 28.0
1_emma_8000_**stereo**, 8.7	1_twocities_8000_**stereo**, 25.2	1_brownell_16000_**stereo**, 41.1
1_emma_8000_**mono**, 9.7	1_twocities_8000_**mono**, 26.7	1_brownell_8000_**stereo**, 45.2
1_thesen	1_choi	1_courtland
1_thesen_16000_**mono**, 21.3	1_choi_16000_**stereo**, 39.6	1_courtland_over40000_**stereo**, 31.8
1_thesen_8000_**mono**, 21.5	1_choi_16000_**mono**, 41.2	1_courtland_over40000_**mono**, 33.0
1_thesen_over40000_**mono**, 21.7	1_choi_over40000_**mono**, 42.7	1_courtland_16000_**mono**, 34.8
1_thesen_8000_**stereo**, 22.2	1_choi_8000_**mono**, 44.1	1_courtland_8000_**mono**, 39.4
1_thesen_16000_**stereo**, 27.3	1_choi_over40000_**stereo**, 88.0	1_courtland_8000_**stereo**, 53.2
1_thesen_over40000_**stereo**, 27.6	1_choi_8000_**stereo**, 88.0	1_courtland_16000_**stereo**, 60.6

Tabelle 5: WER-Werte gruppiert nach Titeln, 6 Beispiele aus dem Korpus

Aus dem Beispiel kann man auch schlussfolgern, dass bei den meisten Titel die beste Transkription bei der Abtastrate von 16000Hz zustande kam. Auch in der vollständigen Liste es ist so: 11 von 18 Titel wurden mit der besten Qualität bei der Abtastrate von 16000Hz transkribiert. Das schlechteste Resultat hingegen liegt im Falle von 15 von 18 Titeln bei 8000Hz.
Es wurde auch kurz untersucht, ob es einen Zusammenhang zwischen dem Sprecher-Geschlecht und der Qualität gibt. Nachdem der komplette Evaluationsprozess durchgeführt wurde, wurde allerdings festgestellt, dass der Korpus zu klein und zu wenig vielfältig ist, um klare Schluss-

folgerungen diesbezüglich zu ziehen. Die Statistiken zu den Geschlechtern werden demzufolge in dem Bericht nicht zusammengefasst.
Das Gleiche gilt auch für den Zusammenhang zwischen der Qualität der Transkriptionen von Interviews, die telefonisch durchgeführt wurden, und der Qualität der Transkriptionen von Gesprächen, die vollständig im gleichen Studio aufgenommen wurden. Daher wird auch auf die Zusammenfassung der Resultate verzichtet.
Zum Abschluss wird noch der vom Tool erreichte minimale WER-Wert mit der State-of-the-Art-Lösung verglichen. Das beste Resultat von dem Tool beläuft sich auf 6.9%, was relativ weit über dem besten Ergebnis von Google (4.9%) liegt.

Diarization-Evaluation

Es wurden insgesamt 78 Transkriptionen von Interviews durchgeführt. Wie früher erwähnt, kommt jedes Interview in 6 Varianten vor (je nach Abtastrate und Anzahl Kanäle). Insgesamt 168 Sprecher treten in der gesamten Kollektion der Aufnahmen auf und erstellen insgesamt 1734 Utterances.
Von den 168 Sprecher wurden durchschnittlich nur 50% erkannt. Mit "erkannt" wird gemeint, dass in den maschinellen Transkriptionen die gleiche Anzahl Sprecher identifiziert wurde, wie im Referenztext angegeben. Somit beträgt die durchschnittliche Speaker Number Error Rate für den gesamten Korpus 0.5.
Wird ein Sprecher nicht erkannt, werden entsprechend auch seine Utterances in der Transkription ausgelassen. Die SNER und die UNER hängen also zusammen. Die durchschnittliche Utterance Number Error Rate beläuft sich auf 0.89. Nur bei zwei Interviews ist sie gleich 0. Bei 36 Transkriptionen ist sie grösser als 0.9. Der durchschnittliche Unterschied zwischen der Anzahl Utterances in der Referenz und in der Hypothese im ganzen Korpus beträgt 7. Der höchste Unterschied beträgt 24 Utterances in drei Transkription vom Interview 3_courtland. Tabelle 6 präsentiert die Ergebnisse von Transkriptionen aller Varianten dieses Interviews aufgeführt.

Referenz: 3_courtland		
Speakers number: 3		
speaker1: **3 utterances** speaker2: **19 utterances** speaker3: **24 utterances**		
3_courtland_over40000_mono	**3_courtland_over40000_stereo**	**3_courtland_16000_mono**
Speakers number: 1 2 speakers not recognized!	Speakers number: 3	Speakers number: 1 2 speakers not recognized!
speaker1: 1 utterance speaker2: - speaker3: - Speaker Number Error Rate: 0.67 Utterance Error Rate: 0.98	speaker1: 2 utterances speaker2: 7 utterances speaker3: 4 utterances Speaker Number Error Rate: 0.0 Utterance Error Rate: 0.72	speaker1: 2 utterances speaker2: - speaker3: - Speaker Number Error Rate: 0.67 Utterance Error Rate: 0.96
3_courtland_16000_stereo	**3_courtland_8000_mono**	**3_courtland_8000_stereo**
Speakers number: 3	Speakers number: 1 2 speakers not recognized!	Speakers number: 3
speaker1: 7 utterances speaker2: 16 utterances speaker3: 20 utterances Speaker Number Error Rate: 0.0 Utterance Error Rate: 0.24	speaker1: 1 utterance speaker2: - speaker3: - Speaker Number Error Rate: 0.67 Utterance Error Rate: 0.98	speaker1: 2 utterances speaker2: 15 utterances speaker3: 18 utterances Speaker Number Error Rate: 0.0 Utterance Error Rate: 0.24

Tabelle 6: Diarization-Statistiken für 3_courtland

Zur Veranschaulichung des Problems wird unten ein Beispiel für falsche Sprechertrennung und falsche Utterance-Erkennung angeführt (siehe Tabelle 7). In der untenstehenden Transkription wird aus drei Utterances nur eine Utterance. Die kurze Äusserung von dem Interviewpartner "Thank you very much" wird in der Transkription völlig ausgelassen.

2_bix.ref	2_bix_over40000_stereo.hyp
It wasn't until the second half of the century that women began to make inroads into engineering programs at US universities and colleges Amy Sue Bix looks at three of these universities—Georgia Tech Caltech and MIT—and how they and their students coped with the arrival of women in her recent book Girls Coming to Tech A History of American Engineering Education for Women She joins us now by phone from her office at Iowa State University where she is the director of the Center for Historical Studies of Technology and Science Amy welcome to the program **(speaker2_1)**	It was until the second half of the 20th century that we began to make inroads into engineering programs at us universities and colleges Amy subic's looks at the three of these universities Georgia Tech Caltech and MIT and how they other students cope with the arrival of women in her recent book girls coming to Tech history of American engineering education for women She joins us now by phone from her office at Iowa State University where she is the director of the center for historical studies of technology and science Amy welcome to the program
Thank you very much (speaker1_1)	**Sprecherwechsel nicht erkannt**
As I mentioned in the introduction historically the gender barrier to women seems to have been especially high in engineering as compared to science and medicine Why was that **(speaker2_2)**	As I mentioned in the introduction historically the gender by Earth Wind seems to be especially high in engineering as compared to science and medicine Why was that **(speaker2_1)**

Tabelle 7: Beispiel für Diarization-Fehler

Die hohen Werte von SNER und UER sind auf einen Fehler im Transkriptionsprozess zurückzuführen. Sobald der Fehler behoben wird, sollte die Evaluation erneut durchgeführt werden.

5.2.2 Manuelle Analyse: Beobachtungen

Im Rahmen von der manuellen Analyse wurden die .sgml-Berichte von sclite untersucht sowie ausgewählte Referenz- und die Hypothesentext verglichen. Dabei wurde Bezug auf die im theoretischen Teil erwähnte Herausforderungen im Bereich Spracherkennung genommen.
Eine der Schwierigkeiten bei der Spracherkennung ist die Intersprechervariabilität: das gleiche Wort wird von verschiedenen Sprechern unterschiedlich ausgesprochen. Es führt zur Entstehung unterschiedlichen Transkriptionen für das gleiche Wort. In der hier untersuchten Kollektion der Transkriptionen wird beispielsweise die Phrase "Techwise Conversations" auf mehrfache Weise transkribiert, was die Tabelle 8 veranschaulicht.

Referenztext	Hypothesentext
techwise conversations	tech my conversations
	tech wise conversations
	the tech wise conversations
	attack wise conversations

Tabelle 8: Beispiele für Intersprechervariabilität

Neben Intersprechervariabilität kommt ein weiteres Phänomen vor: Intrasprechervariabilität. Der gleiche Sprecher kann das gleiche Wort unterschiedlich aussprechen, was zu unterschiedlichen

Transkriptionen führen kann. Ein Beispiel dafür sind die Transkriptionen vom Nachnamen eines IEEE Interviewers Stephen Cass, der sich selbst am Anfang jedes Interviews vorstellt.

Referenztext	Hypothesentext
Cass	Cast
	Casper
	Kass
	Castro
	Katz

Tabelle 9: Beispiele für Intrasprechervariabilität

In keinem Hypothesentext wurde der Nachname von Cass richtig transkribiert. Die Transkription von Namen ist allerdings eine der grösseren Herausforderungen im Bereich Spracherkennung. Auch dafür kann man zahlreiche Beispiele finden:

Referenztext	Hypothesentext
Bergmayer	Burgmeier
Kepler	Coupler
Erik Petigura	Eric Pettigrew
Kodak	Collect
Maria Mitchell	memory of Mitchell

Tabelle 10: Beispiele für fehlerhafte Namen-Transkriptionen

Im theoretischen Teil der Arbeit wurde erwähnt, dass es schwierig ist, aus einer Lautenfolge die richtigen Wörter zu erschliessen, denn Wortgrenzen werden eher selten als Sprechpausen realisiert und demzufolge bestehen oft mehrere Varianten, die Laute in Wörter zusammenzufügen. Diese Erscheinung wird als Fehlsegmentierung bezeichnet [6]. In den Transkriptionen können viele Beispiele für dieses Phänomen gefunden werden.

Referenztext	Hypothesentext
and said	instead
red nose	randalls
which are	witcher
stare at	sterrett
habitable zone	hot ozone

Tabelle 11: Beispiele für Fehlsegmentierung

Eine weitere Schwierigkeit bilden Homophone. In jeder untersuchten Transkription kommen mehrere Beispiele für Transkriptionsfehler, die sich aus der gleichen Aussprache von Wörtern mit verschiedenen Bedeutungen ergeben.

Referenztext	Hypothesentext
were	where
their	there
hart	heart
no	know
too	to
four	for

Tabelle 12: Beispiele für fehlerhafte Transkriptionen bei Homophonen

Anhand der manuellen Analyse wurde auch beobachtet, dass nicht korrekt erkannte Interpunktionszeichen die Auswertung wesentlich beeinflussen. Das sclite Tool fügt die Wörter mit nachfolgenden Interpunktionszeichen zusammen und betrachtet beispielsweise "Hi, " und "Hi" als Substitution. Aus diesem Grund wurden die Interpunktionszeichen aus den Referenz- und Hypothesentexten entfernt, wodurch die WER deutlich reduziert wurde. Die Zeichen wie Apostrophe und Bindestrich wurden hingegen gelassen, da sie Teil von Wörtern bilden.

5.3 Fazit

Im Rahmen der Evaluation wurden die im Tool implementierte Spracherkennung sowie Sprecher-Diarization bewertet. Es wurden dazu drei Messgrössen WER, SNER und UNER berechnet. Die Statistiken und die manuelle Analyse der Berichte und Transkriptionen erlaubten es, die Verbesserungsmöglichkeiten des Tools zu identifizieren und die Herausforderungen von Speech Recognition und Speaker Diarization besser zu verstehen. Die Evaluation ermöglichte es auch, die von dem Tool erreichte Qualität der Spracherkennung mit dem aktuellen State-of-the-art Ergebnis zu vergleichen. Die Resultate der Evaluation werden für die Weiterentwicklung des Tools als Benchmark verwendet.

6 Diskussion und Ausblick

Das Ziel dieser Arbeit war es, ein System für die Interview Transkription zu entwickeln. Die geplante Funktionalität umfasste die Übergabe einer Audio-Datei an die Anwendung über eine grafische Schnittstelle, die Durchführung der Transkription und die Ausgabe des transkribierten Textes in der grafischen Schnittstelle. Mit der Transkription war einerseits reine Speech-to-text-Funktionalität, andererseits auch Sprecher-Diarization gemeint. Für die erste Entwicklungsphase wurde die englische Sprache ausgewählt, weil das Angebot an Transkriptions-Anwendungen für Englisch am umfangreichsten ist und sich die vorhandenen Tools durch die beste Leistung kennzeichnen.

Die am Anfang des Projektes festgelegte Grundfunktionalitäten wurden implementiert. Es wurde auch eine Replay-Funktion entwickelt, welche es ermöglicht, die im Replay-Modus gerade ausgesprochenen Phrase oder Aussage im transkribierten Text hervorzugehoben. Es besteht auch eine Möglichkeit, durch Anklicken eines beliebigen Textfragments das Abspielen des ausgewählten Fragments zu starten. Neben der Implementierung der Transkriptions-Funktionen, wurde auch eine Evaluation der Anwendung durchgeführt. Die Evaluation bezog sich auf die Leistung der Anwendung im Bereich Speech-to-Text und Sprecher-Diarization. Die Bewertung wurde auf einem Korpus von 108 Audio-Dateien mit manuellen Transkriptionen durchgeführt.

Die grösste Herausforderung bei dem Projekt war die Einarbeitung in das Gebiet der Spracherkennung. Wegen geringen Vorkenntnissen des Teams wurde damit am Anfang viel Zeit verbracht. Auch das Auffinden von einem geeigneten Testdatensatz sowie die Zusammenstellung des Speech-Korpus für die Evaluation waren aufwendige Prozesse, die viel Zeit in Anspruch genommen haben.

Die Arbeit an der Anwendung verlief sonst reibungslos. In jeder Iteration wurden die gesetzten Ziele erreicht. Die Planung des Projektes könnte allerdings verbessert werden, indem mehr Tests und mehr Zeit für die Fehlerbehebung in den ersten Iterationen eingeplant würden. Es fehlte anfangs ein umfangreicher Testdatensatz, der geeignete Tests ermöglichen würden. Dadurch haben sich zu Beginn Fehler eingeschlichen, die bei anfangs verfügbaren Audio-Dateien nicht auftauchten, und welche später Probleme bereitet haben. Dies ist in den Evaluationsresultaten im Kapitel 5 ersichtlich. Es handelt sich vor allem um einen Fehler, bei dem grössere Text-Segmente in dem transkribierten Text ausgelassen wurden und um ein Problem mit der richtigen Sprecherwechsel-Erkennung. Die Durchführung der Evaluation könnte auch optimiert werden. Wahrscheinlich wären Python-Open-Source-Tools für WER- und DER-Berechnung für die Zwecke des Projektes eine bessere Lösung. Sie müssten zwar an die Projektbedürfnisse noch angepasst werden, aber möglicherweise wäre die Durchführung der Evaluation dadurch effizienter. Ausserdem würde die Evaluation idealerweise auf dem Switchboard Korpus oder einem anderen für die Evaluationszwecke häufig eingesetzten Korpus durchgeführt. Letztens wurde auch die Schlussfolgerung gezogen, dass es einfacher gewesen wäre, den Python Teil auf einem zentralen Webserver verfügbar zu machen, worauf alle Electron Clients zugreifen könnten.

Einige dieser Lessons Learned stellen ein wichtiges Input für die weitere Entwicklung der Anwendung dar. Im Laufe des Projektes wurden mehrere Funktionalitäten identifiziert, die in den weiteren Entwicklungsphasen implementiert werden könnten (vgl. Product Backlog im Anhang).

Eine der wichtigsten Anforderungen wäre die Implementierung der Transkription von deutschen Audio-Dateien. Idealerweise könnte die Anwendung auch den schweizerischen Dialekt unterstützen. Es gibt auf dem Markt noch keine Transkriptionsanwendung, die Transkriptionen von Schweizerdeutsch anbietet.
Es wäre ausserdem interessant eine weitere umfangreichere Evaluation durchzuführen, in der die Diarization mit der Standard-Messgrösse DER bewertet werden könnte. Dazu müssten manuelle Transkriptionen zur Verfügung stehen, wo die Anfangs- und Endzeit jeder Sprecher-Utterance angegeben ist. Die Evaluation könnte auch durch verschiedene WER-Berechnungsvarianten erweitert werden und so könnte beispielsweise die Auswirkung der Entfernung von Interpunktionszeichen untersucht werden.
Darüber hinaus könnte untersucht werden, welche STT Fehler mit einem verbesserten Sprachmodell vermieden oder durch Postprocessing der Transkription behoben werden könnten.
Es wäre auch erwägenswert, andere APIs, wie beispielsweise diejenige von IBM, auszuprobieren oder eventuell als Alternative zur Google-Lösung anzubieten.
Es besteht ein grosses Potential für Erweiterungen und Optimierungen der oben beschriebenen Transkriptions-Anwendung. Mit den Erkenntnissen aus der bisherigen Entwicklungsphase, durch die Verfügbarkeit von verschiedenen STT-Technologien sowie dank zahlreicher und umfangreicher Untersuchungen zum Thema Spracherkennung und Sprecher-Diarization, die online nachvollzogen werden können, kann sich die Anwendung zu einem Tool entwickeln, das einerseits den Benutzern gute Qualität von STT und andererseits viele benutzerfreundliche Funktionalitäten anbietet.

7 Verzeichnisse

7.1 Literaturverzeichnis

[1]	M. González, J. Moreno, J.L. Martínez, P. Martínez, "An Illustrated Methodology for Evaluating ASR Systems". In: Detyniecki M., García-Serrano A., Nürnberger A., Stober S. (eds), "Adaptive Multimedia Retrieval. Large-Scale Multimedia Retrieval and Evaluation." 2011, Lecture Notes in Computer Science, vol 7836. Springer, Berlin, Heidelberg, S. 33–42
[2]	o.V. *sclite Dokumentation* [Online]. URL: http://www1.icsi.berkeley.edu/Speech/docs/sctk-1.2/sclite.htm [Stand: 20.12.2018]
[3]	B.Pfister, T.Kaufmann, "Sprachverarbeitung, Grundlagen und Methoden der Sprachsynthese und Spracherkennung", Springer Vieweg, 2017, S. 22-28
[4]	D. Kolossa, "Grundlagen der automatischen Spracherkennung", Institut für Kommunikationsakustik, Ruhr-Universität Bochum, 21. April 2017, S. 73-86
[5]	M. Branscombe (2017). *Beyond the Switchboard: The Current State of the Art in Speech Recognition* [Online]. URL: https://thenewstack.io/speech-recognition-getting-smarterstate-art-speech-recognition
[6]	S. Clematide, L. Nigg, "Vorlesungsskript. Einführung in die Computerlinguistik I", Institut für Computerlinguistik, Universität Zürich, 27. März 2007 https://files.ifi.uzh.ch/cl/siclemat/lehre/ws0607/ecl1/script/html/script3.html#scriptch9.html
[7]	T. Stadelmann, Class Lecture, Topic: "Artificial Intelligence V10: Probabilistic Learning" School of Engineering, Zurich University of Applied Sciences, Winterthur, ZH, 4.12.2018
[8]	X. Anguera, *Model-Based Segmentation* [Online]. URL: http://www.xavieranguera.com/phdthesis/node15.html [Stand: 15.12.2018]
[9]	X. Anguera, *Metric-Based Segmentation* [Online]. URL: http://www.xavieranguera.com/phdthesis/node12.html [Stand: 15.12.2018]
[10]	I. Lapidot, H. Guterman und A. Cohen, "Unsupervised Speaker Recognition Based on Competition Between Self-Organizing Map" in *IEEE Transactions on Neural Networks*, Vol. 13, No. 4, Jul. 2002 (link to delete later: http://www.ee.bgu.ac.il/~itsik/papers/IEEE_Tran_NN.pdf)
[11]	X. Anguera, *Diarization Error Rate* [Online]. URL: http://www.xavieranguera.com/phdthesis/node108.html [Stand: 16.12.2018]
[12]	A. Zhang, Q. Wang, Z. Zhu, J. Paisley, C. Wang, *Fully Supervised Speaker Diarization.* arXiv preprint arXiv:1810.04719, 2018

[13]	Nuance Communication Inc. *Dragon Speech Recognition – Get More Done by Voice | Nuance* [Online]. URL: https://www.nuance.com/dragon.html [Stand: 16.12.2018]
[14]	TranscribeMe Inc. *Transcription Rates and Pricing – TranscribeMe* [Online]. URL: https://www.transcribeme.com/pricing [Stand: 16.12.2018]
[15]	Avignon Université. *Alizé opensource speaker recognition* [Online]. URL: https://alize.univ-avignon.fr/ [Stand: 27.12.2018]
[14]	M. Kotti, V. Moschou, C. Kotropoulos. (31.10.2007). *Speaker Segmentation and Clustering* [Online]. URL: http://poseidon.csd.auth.gr/papers/PUBLISHED/JOURNAL/pdf/Kotti08a.pdf [Stand: 17.11.2018]
[15]	What-When-How. *Audio Features (Audio Processing) (Video Search Engines)* [Online]. URL: http://what-when-how.com/video-search-engines/audio-features-audio-processing-video-search-engines/ [Stand: 15.12.2018]
[16]	T. Kemp, M. Schmidt, M. Westphal, A. Waibel. *Strategies for automatic segmentation of audio data* [Online]. URL: https://www.ee.columbia.edu/~dpwe/papers/KempSWW00-audseg.pdf [Stand: 15.12.2018]
[17]	S. Hooja. (03.11.2018). *Why Python is the Right Programming Language for Data Science* [Online]. URL: https://datafloq.com/read/why-python-programming-language-data-science/2426 [Stand 19.12.2018]
[18]	A. Ronacher. *Flask – web development, one drop at a time* [Online]. URL: http://flask.pocoo.org [Stand 20.12.2018]
[19]	*Electron* [Online]. URL: https://electronjs.org [Stand 20.12.2018]
[20]	*Vue.js* [Online]. URL: https://vuejs.org [Stand 20.12.2018]
[21]	*PyInstaller* [Online]. URL: https://www.pyinstaller.org [Stand 20.12.2018]
[22]	E.G. Schukat-Talamazzini, "Automatische Spracherkennung. Statistische Verfahren der Musteranalyse", Vieweg Verlag, 1995, S. 45-74

[23]	A. Hallerbach, "HMM/KNN zur Spracherkennung", Universität Ulm, 2005, S. 3-6
[24]	K.U. Carstensen, Ch. Ebert, C. Ebert, S. Jekat, H. Langer, R. Klabunde, " Computerlinguistik und Sprachtechnologie: Eine Einführung", Springer-Verlag, 04.11.2009, S. 218-220
[25]	K. Schulz, Ch. Ringlstetter, F. Schiel, "Sprachmodelle", Universität München, 2006

8 Anhang

8.1 Projektmanagement

8.1.1 Offizielle Aufgabenstellung, Projektauftrag

Aufgrund der kurzfristen Themenbestimmung ist keine offizielle Aufgabenstellung gegeben. Die Ziele wurden bilateral zwischen Prof. Dr. Mark Cieliebak und den Studierenden vereinbart (vgl. Abschnitt 1.2).

8.1.2 Projektplan

SW6 (23.10 - 29.10) "Projekt-Setup"

- Anforderungen als User Stories dokumentieren
- SW6 Tasks
- **KEIN DEMO**

SW7 (30.10 - 5.11) "Transkription im Backend: Grundfunktionen"

- Backend für US1
- SW7 Tasks
- **KEIN DEMO**

SW8 (6.11 - 12.11) "Transkription mit Frontend: Grundfunktionen"

- US1 fertig stellen
- SW8 Tasks
- **DEMO**: Englische Audio-Dateien werden über ein GUI hochgeladen und anschliessend transkribiert. Der transkribierte Text (ohne Sprecher IDs) wird im Frontend angezeigt.

SW9 (13.11 - 19.11) "Diarization und Sprecher-Benennen"

- US2 fertig stellen
- US7 fertig stellen
- SW9 Tasks
- **DEMO**: Englische Audio-Dateien werden über ein GUI hochgeladen und anschliessend transkribiert. Der transkribierte Text mit Sprecher IDs wird im Frontend angezeigt. Der Benutzer kann die Sprecher benennen. Im transkribierten Text stehen dann Sprechernamen anstatt Sprecher IDs.

SW10 (20.11 - 26.11) "Replay-Modus 1"

- US3 fertig stellen
- US6 fertig stellen
- US8 fertig stellen
- SW10 Tasks
- **KEIN DEMO**

SW11 (27.11 - 3.12) "Replay-Modus 2"

- SW11 Tasks

- **DEMO**: Dem Benutzer steht die Replay-Funktion zur Verfügung. Beim Abspielen werden die entsprechenden Fragmente im transkribierten Text hervorgehoben. Durch Anklicken eines beliebigen Textfragments wird das Abspielen des Fragments gestartet.

SW12 (4.12 - 10.12) "Bug fixes, Bericht schreiben"

- SW12 Tasks
- Bericht schreiben

SW13 (11.12 - 17.12) "Bug fixes, cleanup, Bericht schreiben"

- SW13 Tasks
- Bericht schreiben

SW14 (18.12 - 21.12) "Bericht fertigstellen"

- SW14 Tasks
- Bericht schreiben

Übrige User Stories

- US20 Interview Export zur Bearbeitung ausserhalb des Tools

8.2 Product Backlog

US6	**Als Interviewer möchte ich einzelne Worte im Transkript editieren können, um Fehler im Transkript zu beheben.**
Akzeptanz Kriterien	• Es müssen einzelne Worte korrigierbar sein.

US7	**Als Interviewer möchte ich die gleichen Worte, welche falsch transkribiert wurden, korrigieren können, damit ich Zeit spare und nicht jedes Wort einzeln korrigieren muss.**
Akzeptanz Kriterien	• Es muss ersichtlich sein, welche Passagen dies betrifft. • Es muss eine Meldung bestätigt werden, um alle gleich falschen Worte zu ändern.

US8	**Als Interviewer möchte ich die Sprecherabschnitte korrigieren, falls diese falsch erkannt wurden, damit das Transkript mit dem Audio übereinstimmt.**
Akzeptanz Kriterien	• Die Sprecherabschnitte müssen ohne grossen Aufwand aktualisiert werden können.

US9	**Als Interviewer möchte ich Worte an einer beliebigen Stelle im Transkript einfügen oder löschen können, damit das Transkript mit dem Audio übereinstimmt.**

Akzeptanz Kriterien	• Die Worte müssen ohne grossen Aufwand hinzugefügt werden können. • Die Worte müssen ohne grossen Aufwand entfernt werden können.

US10	**Als Interviewer möchte ich das Transkript exportieren und lokal speichern können, um ausserhalb der Anwendung weiter daran zu arbeiten.**
Akzeptanz Kriterien	• Das Export-Format (docx, txt, html, etc.) muss ausgewählt werden können.

US11	**Als Interviewer möchte ich das komplette Projekt speichern und wieder öffnen können, damit über längere Zeit am gleichen Transkript gearbeitet werden kann.**
Akzeptanz Kriterien	• Das Projekt muss gespeichert werden können. • Das Projekt muss geöffnet werden können. • Bei Änderungen am Projekt soll nach dem ersten Speichern automatisch gespeichert werden können.
US12	**Als Interviewer möchte ich Änderungen, die ich an der Transkription vogenommen habe, rückgängig machen.**

US13	**Als Interviewer möchte ich gleichzeitig alle gleichen Wortvorkommen anpassen.**

US13	**Als Interviewer möchte ich, dass die Sprache von dem hochgeladenen Interview automatisch erkannt wird.**

US14	**Als Interviewer möchte ich, dass Anzahl Sprecher in dem hochgeladenen Interview automatisch erkannt wird.**

US15	**Als Interviewer möchte ich, dass die Originalwörter, die ich korrigiert habe, für langfristiges Lernen von verbesserten Wörtern gespeichert werden.**

US16	**Als Interviewer möchte ich, die Originaltranskription mit der maschinellen vergleichen, um die Fehler zu erkennen und die Leistung der Anwendung zu evaluieren.**

US17	**Als Interviewer möchte ich, dass die Sprecher-Aussagen entsprechend farbig markiert werden, damit die Unterscheidung deutlicher wird.**

US18	**Als Interviewer möchte ich, dass Wörter nach Confidence in verschiedenen Farbe markiert werden.**

US19	**Als Interviewer möchte ich, dass die deutsche Sprache unterstützt wird.**

US19	**Als Interviewer möchte ich, dass auch Videos über das Tool transkribiert werden können.**

US20	**Als Interviewer möchte ich, dass das Tool auch als Smartphone App (iOS, Android) verfügbar ist.**

US20	**Als Interviewer möchte ich, dass mehr Audioformate unterstützt sind. (Eventuell werden sie von der Anwendung konvertiert.)**

US21	**Als Interviewer möchte ich, dass schweizerische Dialekte zu Hochdeutsch transkribiert werden könnten.**

8.3 Technische Dokumentation

8.3.1 Anleitung zur Einrichtung der Google Cloud Umgebung

Hinweis: Gilt für Spyder, bei PyCharm ist die Einrichtung etwas einfacher

https://console.cloud.google.com/freetrial

1. Google Cloud Konto erstellen

https://cloud.google.com/text-to-speech/docs/quickstart-client-libraries

2. GCP-Projekt in der Google Cloud Konsole erstellen
3. Abrechung fürs Projekt in der Google Cloud Konsole aktivieren

https://cloud.google.com/speech-to-text/docs/quickstart-client-libraries

4. Google Cloud API aktivieren
5. Authentifizierung in der Google Cloud Konsole einrichten (Dienstkonto erstellen)
6. Umgebungsvariable auf den Pfad zum Dienstkontoschlüssel festlegen

https://cloud.google.com/sdk/docs/

7. Google Cloud SDK installieren

https://cloud.google.com/python/setup#installing_and_using_virtualenv:

8. Python-Entwicklungsumgebung einrichten (Python2 und Python3 benötigt)
9. Python-Projekt erstellen
10. Virtualenv installieren
11. Virtualenv-Umgebung im Projekt erstellen
12. Virtualenv-Tool aktivieren

https://cloud.google.com/text-to-speech/docs/quickstart-client-libraries

13. Google Cloud Clientbibliothek installieren
14. Beispiel-Anfrage zur Audiotranskription erstellen

8.3.2 Anleitung zur Einrichtung der Entwicklungsumgebung

Interview Transcription

The project code is located in `/src/interview-transcription` so you'll need to navigate in there first.

SETUP

Installation:

- Install Python2 and Python3
- setup a new project under `/src/interview-transcription` with a virtual environment https://cloud.google.com/python/setup#installing_and_using_virtualenv
- paste google json file to root directory `/src/interview-transcription`. There is no need to set an environment variable.
- activate virtualenv and install dependencies:

```
source venv/Scripts/activate (OSX: source env/bin/activate)

pip install -r requirements.txt
npm install

pyinstaller backend/api.py --distpath backend-dist --clean && rm -rf
 build/api
```

- edit pathex in api.spec

RUN DEVELOPMENT

application has to be started in virtual environment

```
source venv/Scripts/activate (OSX: source env/bin/activate)

npm run dev
```

If error `grpc module not found` or similar occurs, try following:

- Install packages `google-cloud-storage` and `google-cloud-speech` one for one, directly in pyCharm or with pip

File Tree:

Here's a brief overview of where what is located (some stuff omitted):

```
-- src/interview-transcription/
    -- backend/ (python application files)
    -- build/ (contains built electron executables and icons files)
    -- src/
        -- main/ (contains electron entry point @ index.js)
```

```
        -- renderer/ (frontend logic / vue framework)
    -- static/ (static resources)
    -- test/ (frontend tests)
```

Debugging:
The python subprocess uses the same stdout and stderr as the main process so you can check for issues in the terminal.

PACKAGE

First check `src/main/index.js` and comment out the development lines for the packaging lines. Then compile your python files:

```
# compile python files
npm run build:python
```

And after package the electron app:

```
# build windows executable electron app
npm run build:win32

# build osx electron app
npm run build:Darwin
```

The electron application is build into `build/interview-transcription-...`
Further build commands can be found in `package.json`.

DEACTIVATE

Deactivate your python virtual environment:

Deactivate

8.3.3 Installationsanleitung für Endbenutzer

Interview-Transkription Order (gezippt) extrahieren, anschliessend Interview-Transkription.exe ausführen.

8.3.4 API Dokumentation (für Frontend-Backend-Kommunikation)

Beschreibung der JSON-API des Backend:

Übersicht:

URI	Beschreibung
`/create`	API für das Erstellen eines neuen Projektes
`/upload`	API für den Upload der Audio Datei
`/transcribe`	API für das Transkribieren einer Audio Datei
`/update`	API für das Aktualisieren einer Transkription
`/save`	API für das Speichern eines Projekts
`/open`	API für das Laden eines Projekts

POST – 200 /rest/create

```
{
    "audio": {
        "uri": ".../testdata/english_conversation_short.wav",
        "metadata": {
```

```
            "encoding": "LINEAR16",
            "sample_rate": 44100,
            "language": "en-US",
            "speaker_count": 1,
            "audio_channel_count": 2,
            "model": "default"
        },
        "content": "UklGRn7vggBXQVZFZm10IBAAAAABAAIARKwAABCx-
                    AgAEABAAZGF0YZTuggABAP///...(base64 encoded)"
    }
}
```

POST - 200 /rest/upload


```
{
    "uri": "gs://interview-transcription/demo-audio-
            files/english_conversation_short.wav"
}
```


POST - 200 /rest/transcribe, /rest/update, /rest/open


```
{
    "interview": {
        "audio": {
            "uri": "gs://interview-transcription/demo-audio-
                    files/english_conversation_short.wav",
            "metadata": {
                "encoding": "LINEAR16",
                "sample_rate": 44100,
                "language": "en-US",
                "speaker_count": 2,
                "audio_channel_count": 2,
                "model": "default"
            },
            "content": "UklGRn7vggBXQVZFZm10IBAAAAABAAIARKwAABCx-
                        AgAEABAAZGF0YZTuggABAP///...(base64 encoded)"
        },
        "transcription": {
             "elements": [
                 {
                     "alternative": "",
                     "confidence": 97.04,
                     "corrected": "",
                     "recognized": "Thank",
                     "time_frame": {
                         "start_seconds": 12,
                         "start_nanos": 200000000,
                         "end_seconds": 13,
                         "end_nanos": 900000000
                     }
                 },
                 {
                     "alternative": "",
                     "confidence": 98.763,
                     "corrected": "",
                     "recognized": "you.",
```

```
            "time_frame": {
                "start_seconds": 13,
                "start_nanos": 600000000,
                "end_seconds": 13,
                "end_nanos": 900000000
            }
        },
        {
            "alternative": "",
            "confidence": 98.763,
            "corrected": "",
            "recognized": "Could",
            "time_frame": {
                "start_seconds": 13,
                "start_nanos": 900000000,
                "end_seconds": 18,
                "end_nanos": 100000000
            }
        }
    ],
    "speakers": [
        {
            "id": 1,
            "name": "Jane",
            "speak_times": [
                {
                    "start_seconds": 0,
                    "start_nanos": 0,
                    "end_seconds": 3,
                    "end_nanos": 400000000
                },
                {
                    "start_seconds": 4,
                    "start_nanos": 900000000,
                    "end_seconds": 9,
                    "end_nanos": 300000000
                }
            ]
        },
        {
            "id": 2,
            "name": "Alex",
            "speak_times": [
                {
                    "start_seconds": 3,
                    "start_nanos": 400000000,
                    "end_seconds": 4,
                    "end_nanos": 900000000
                },
                {
                    "start_seconds": 9,
                    "start_nanos": 300000000,
                    "end_seconds": 12,
                    "end_nanos": 200000000
```

```
            }
        ]
    }
],
"speaker_frames": [
    {
        "speaker_id": 1,
        "speaker_name": "Jane",
        "elements": [
            {
                "alternative": "",
                "confidence": 0,
                "corrected": "",
                "recognized": "Excuse",
                "time_frame": {
                    "start_seconds": 1,
                    "start_nanos": 100000000,
                    "end_seconds": 2,
                    "end_nanos": 0
                }
            },
            {
                "alternative": "",
                "confidence": 98.763,
                "corrected": "",
                "recognized": "me.",
                "time_frame": {
                    "start_seconds": 2,
                    "start_nanos": 0,
                    "end_seconds": 2,
                    "end_nanos": 200000000
                }
            }
        ],
        "time_frame": {
            "start_seconds": 0,
            "start_nanos": 0,
            "end_seconds": 3,
            "end_nanos": 400000000
        }
    },
    {
        "speaker_id": 2,
        "speaker_name": "Alex",
        "elements": [
            {
                "alternative": "",
                "confidence": 96.038,
                "corrected": "",
                "recognized": "please?",
                "time_frame": {
                    "start_seconds": 3,
                    "start_nanos": 400000000,
                    "end_seconds": 4,
```

```
                                "end_nanos": 900000000
                            }
                        }
                    ],
                    "time_frame": {
                        "start_seconds": 3,
                        "start_nanos": 400000000,
                        "end_seconds": 4,
                        "end_nanos": 900000000
                    }
                }
            ]
        }
    }
}
```


POST - 200 /rest/save


```
{
    "path": ".../testdata/english_conversation_short.intts"
}
```


8.4 Speech-Korpus: Quellen und Originaltitel

- Quellen von Audiobüchern und eBooks: http://www.openculture.com/freeaudiobooks und http://www.openculture.com/free_ebooks
- Quellen von Interviews: https://spectrum.ieee.org

Dateipräfix im Speech-Korpus	Originaltitel und Verfasser
Audiobücher	
1_emma	"Emma" von Jane Austen
1_roomview	"A Room with a view" von E. M. Forster
1_sherlock	"Sherlock Holmes" von Arthur Conan Doyle
1_heartdarkness	"Heart of Darkness" von Joseph Conrad
1_twocities	"A Tale of Two Cities" von Charles Dickens
IEEE Podcasts	
1_bergmayer /_2_bergmayer	"Is the End of Net Neutrality Near?": Interview mit John Bergmayer
1_bix / 2_bix	"How Women Got Through the Doors of Engineering Schools": Interview mit Amy Sue Bix
1_brownell / 2_brownell	"D-Wave Aims to Bring Quantum Computing to the Cloud": Interview mit Vern Brownell
1_ brynjolfsson / 2_brynjolfsson	"The Second Machine Age: Avoiding the Dark Side of the Digital Revolution": Interview mit Erik Brynjolfsson
1_carroll / 2_carroll	The Reality Behind "Halt and Catch Fire": Interview mit Paul Carroll
1_choi / 2_choi	"Identifying the Recipe for Literary Success": Interview mit Yejin Choi

1_gellings / 2_gellings	"Clark Gellings: The Future of the Power Grid": Interview mit Clark Gellings
1_harris / 2_harris	"Who Determines the Value of Patents?": Interview mit Mark Harris
1_kress / 2_kress	"Nancy Kress: How Science Fiction Helps Us Rehearse for the Future": Interview mit Nancy Kress
1_thesen / 2_thesen	"EV Evangelism Starts at Home": Interview mit Sven Thesen
1_westgate / 2_westgate	"Electric Shocks Preferred to Thinking (Especially by Men)": Interview mit Erin Westgate
1_courtland / 3_courtland	"Kepler Is Dead. Long Live Kepler": Interview mit Rachel Courtland
1_ strickland / 3_strickland	"Nick Bostrom Says We Should Trust Our Future Robot Overlords": Interview mit Eliza Strickland

8.5 Evaluationsszenarien

#	Szenario-ID	Sprecher-Anzahl	Geschlecht	Störungen	Abtastrate	Anzahl Kanäle
1	S1.1	1	weiblich	ohne	8000	Mono
2	S1.2					Stereo
3	S1.3				16000	Mono
4	S1.4					Stereo
5	S1.5				über 40000	Mono
6	S1.6					Stereo
7	S1.7		männlich		8000	Mono
8	S1.8					Stereo
9	S1.9				16000	Mono
10	S1.10					Stereo
11	S1.11				über 40000	Mono
12	S1.12					Stereo
13	S2.1	2	weiblich und männlich gemischt	ohne	8000	Mono
14	S2.2					Stereo
15	S2.3				16000	Mono
16	S2.4					Stereo
17	S2.5				über 40000	Mono
18	S2.6					Stereo
19	S2.7			mit	8000	Mono
20	S2.8					Stereo
21	S2.9				16000	Mono
22	S2.10					Stereo
23	S2.11				über 40000	Mono
24	S2.12					Stereo
25	S2.13	2	weiblich und männlich gemischt	ohne	8000	Mono
26	S2.14					Stereo
27	S2.15				16000	Mono
28	S2.16					Stereo
29	S2.17				über 40000	Mono
30	S2.18					Stereo
31	S2.19			mit	8000	Mono
32	S2.20					Stereo

33	S2.21				16000	Mono
34	S2.22					Stereo
35	S2.23				über 40000	Mono
36	S2.24					Stereo
37	S3.1	3	weiblich und männlich gemischt	ohne	8000	Mono
38	S3.2					Stereo
39	S3.3				16000	Mono
40	S3.4					Stereo
41	S3.5				über 40000	Mono
42	S3.6					Stereo
43	S3.7			mit	8000	Mono
44	S3.8					Stereo
45	S3.9				16000	Mono
46	S3.10					Stereo
47	S3.11				über 40000	Mono
48	S3.12					Stereo

8.6 Evaluationsautomatisierung

Alle Evaluationsschritte werden über einen Shell-Skript `sclite_evaluation.ksh` ausgeführt. Das Skript nimmt folgende Parameter als Input:

- ref_dir: Ordner, wo sich die Referenztranskriptionen befinden
- hyp_dir: Ordner, wo sich die Transkriptionen der Interview-Transkriptionsanwendung befinden
- wav_dir: Ordner, wo sich der ganze Speech-Korpus befindet
- eval_dir: Ordner, wo die Statistiken gespeichert werden sollten

Die einzelnen Schritte werden unten detailliert beschrieben.

Um die Evaluation mit sclite durchzuführen, muss zuerst das sclite Tool installiert werden. Die Einleitung zur Installation der Anwendung befindet sich unter folgender Adresse: https://github.com/usnistgov/SCTK/blob/master/README.md.

8.6.1 Input-Dateien-Preprocessing

Manuelle Erstellung von Referenztranskriptionen

Zuerst muss die Originaltranskription der Audiodatei erstellt werden (wird in der Regel vom Internet heruntergeladen). --> Der transkribierte Text wird als ..._raw.ref-Datei gespeichert. Bei Audios mit mehreren Sprechern ist zu beachten, dass jede Äusserung eines Sprechers auf einer neuen Zeile angefangen muss und es muss am Anfang der Zeile der Name des Sprechers stehen mit Doppelpunkt (z.B. 'John Cass:')

Preprocessing von Inputdateien (evaluation_preprocessing.py)

- .ref-Dateien erstellen
  Die manuell erstellte raw.ref-Datei wird in eine .ref-Datei mit entsprechender Formatierung konvertiert.
- .hyp-Dateien erstellen
  Zuerst wird die gegebene Audiodatei mit der Interview-Transkriptionsanwendung transkribiert.
  Der transkribierte Text wird anschliessend formatiert, um die Anforderungen des sclite Tools zu erfüllen.

Transkriptionen mit mehreren Sprechern in Transkriptionen mit einem Sprecher umwandeln
Damit auch Interviews mit einer fehlerhaften Diarization hinsichtlich der STT-Qualität untersucht und evaluiert werden können, werden die Interview-Transkriptionen mit mehreren Sprechern in eine Transkription mit einem Sprecher umgewandelt.

8.6.2 Evaluation

Die Evaluation mit sclite besteht im Vergleich der .ref-Dateien ("Reference") mit den .hyp-Dateien ("Hypothesis") (der sogenannte Alignment-Prozess).

Es werden 2 Schritte in sclite ausgeführt:

- Erstellung von der .sys-Datei mit Statistiken
  `asclite -h 'hyp_file_name' -r 'ref_file_name' -i spu_id -o sum`
- Erstellung von der .sgml-Datei mit den Details des Alignments
  `asclite -h 'hyp_file_name' -r 'ref_file_name' -i spu_id -o sgml`

8.6.3 Statistiken

Die Statistiken werden ausserhalb von sclite generiert:

- Erstellung von Statistiken bezüglich Diarization
  Die Statistiken werden als `/statistics/diarization_eval_stats.res gespeichert`.
- Erstellung von Statistiken bezüglich WER, Insertions, Deletions, Substitutions
  Die Statistiken werden als `/statistics/stt_eval_stats.res gespeichert`.

8.7 USB-Stick

Dem Bericht wird ein USB-Stick angehängt. Er enthält folgende Ordner:

- interview-transcription: enthält die ganze Codebase von der Anwendung
- sclite_evalution: enthält den Speech-Korpus, die Input- und Output-Dateien von sclite Tool sowie Statistiken
- Bericht im PDF-Format