Michael Paierl

Detektion und Visualisierung von F0-Berechnungsfehlern

Bachelorarbeit

eingereicht an der

Technischen Universität Graz

Betreuerinnen

Saskia Wepner
Anneliese Kelterer

Institut für Signalverarbeitung und Sprachkommunikation

Graz, September, 2021

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Bachelorarbeit identisch.

Graz, am

(Unterschrift)

Acknowledgements

Die Durchführung dieser Bachelorarbeit wäre ohne die Hilfe engagierter Menschen nicht möglich gewesen.
An erster Stelle möchte ich mich bei meinen Betreuerinnen Saskia Wepner und Anneliese Kelterer bedanken. Sie haben mich tatkräftig unterstützt und mir die Möglichkeit gegeben, diese Bachelorarbeit zu schreiben. Durch ihr Fachwissen auf den Gebieten der Signalverarbeitung sowie Linguistik und Phonetik konnten alle aufkommenden Fragen sofort geklärt werden. Mit dieser Arbeit bekam ich die Gelegenheit, mich in verschiedene Konzepte der Signalverarbeitung und Phonetik zu vertiefen. Außerdem musste ich ein verständliches Dokument erstellen, bei dem ich große Unterstützung von meinen Betreuerinnen bekam.
Der größte Dank gilt meinen Eltern. Vielen Dank für die finanzielle Unterstützung sowie Euren motivierenden Beistand während meines gesamten Studiums.

Abstract (German)

Die genaue Betrachtung der Grundfrequenzkurve ist ein essenzieller Teil der Sprachsignalanalyse. Es gibt einige Algorithmen, um die Grundfrequenz eines Sprachsignals zu bestimmen. Häufige Probleme sind Berechnungsfehler, wie Frequenzsprünge und andere Artefakte, die das Sprachsignal nicht korrekt repräsentieren. Manche Algorithmen versuchen, diese Unstetigkeiten zu finden und automatisch zu korrigieren. Diese Korrekturen sind für die Anwendungen in der Sprachanalyse oft nicht von ausreichender Genauigkeit. Für die prosodische Analyse ist es jedoch essenziell, Fehlkorrekturen der F0 zu vermeiden, die zu einer verzerrten Darstellung eines der wichtigsten akustischen Parameter führen würden, die in diesen Untersuchungen von Interesse sind. Daher müssen wir mögliche unerwünschte F0-Tracking-Fehler erkennen und für linguistische Expert:innen verarbeitbar machen, anstatt automatische Korrekturen durchzuführen. Das Ziel dieser Arbeit ist es, ein Werkzeug zu implementieren, das mögliche unerwünschte Diskontinuitäten in F0-Kurven von Sprachsignalen erkennt und Korrekturvorschläge zur manuellen Evaluierung visualisiert.

Abstract

Tracking of the fundamental frequency is an essential part of the analysis of speech signals. There are many algorithms out there for determining the fundamental frequency of speech signals. Common issues are tracking errors, such as octave-jumps and other artefacts which do not accurately represent the speech signal. Some algorithms try to detect such unwanted discontinuities and perform automatic corrections. Sometimes, such corrections are not of sufficient precision. For prosodic analysis, however, it is essential to avoid miscorrections of F0 that would lead to a distorted representation of one of the main acoustic correlates of interest in these investigations. Hence, we need to detect possible unwanted F0 tracking errors and make them processable for linguist experts instead of doing automatic corrections. The aim of this thesis is to implement a tool to detect, cluster and visualise possible unwanted discontinuities in F0 curves of speech signals.

Inhaltsverzeichnis

1 Einleitung

Bei zwischenmenschlicher Kommunikation ist die Bedeutung des Gesprochenen sehr wichtig. Um diese mittels Sprachanalyse richtig interpretieren zu können, müssen einige Parameter berücksichtigt werden. Für die alltägliche Kommunikation ist es nicht nur wichtig, was gesagt wird, sondern auch wie etwas gesagt wird. Wie etwas gesagt wird, was im Zusammenhang auch die Bedeutung eines Satzes verändern kann, wird in der Prosodie, einem Teilgebiet der Phonetik, genauer betrachtet. Die Prosodie beschreibt die Sprachmelodie, Rhythmik, Betonung und Klangfarbe von Sprache, und fasst Eigenschaften von Sprache zusammen, die sich nicht auf einzelne Laute, sondern lautüberspannende Einheiten, beziehen [1]. Die Sprachmelodie wird durch die Grundfrequenz (F0) der menschlichen Stimme beschrieben und die richtige Berechnung dieser ist essenziell für eine prosodische Analyse.
Die menschliche Stimme ist ein aus Grundschwingung und mehreren Oberschwingungen (Formanten) bestehendes periodisches Signal. Die Grundschwingung wird durch die Grundfrequenz, die sogenannte „F0“, in Hz (Schwingungen pro Sekunde) beschrieben. Sie ist die tiefste Frequenz aller im Signal vorkommenden Schwingungen. Bei der menschlichen Stimme ist es jedoch so, dass die Schwingung nicht perfekt periodisch, sondern quasi-periodisch ist. Nicht bei allem Gesprochenen ist eine F0 vorhanden. Es gibt stimmhafte sowie stimmlose Laute. Bei stimmhaften Lauten vibrieren die Stimmlippen, wodurch eine Periodizität zu erkennen ist und daraus folgend auch eine F0 vorhanden ist. Bei stimmlosen Lauten ist die Geräuschquelle eine andere, z.B. Turbulenzen im Mundraum bei einem [s], und die Stimmlippen schwingen nicht. Daher haben stimmlose Laute auch keine F0. Stimmhafte Laute sind Vokale (z.B. [a], [o]) und bestimmte Konsonanten, sogenannte Sonoranten (z.B. [ɾ], [l], [m]) und stimmhafte Plosive und Frikative (z.B. französisch <d> in deux „zwei“ und <j> in jour „Tag“). Beispiele für stimmlose Laute sind die Plosive [p] und [t], und die Frikative [f] und [s] [2]. Falls bei stimmlosen Lauten eine F0 detektiert wird, kann davon ausgegangen werden, dass es sich hierbei um einen Berechnungsfehler handelt.
Die Grundfrequenz unterscheidet sich stark zwischen den Geschlechtern. Bei Männern liegt die durchschnittliche Grundfrequenz bei etwa 120 Hz, bei Frauen bei etwa 200 Hz [3].
Der Stimmumfang, d.h. der Frequenzbereich in dem die sich die F0 bewegt, ist eine wichtige Eigenschaft der menschlichen Stimme. Er ist von Mensch zu Mensch verschieden, und abhängig von der Betrachtungsweise zeigt der Stimmumfang große Unterschiede zwischen Geschlechtern. Betrachtet man ihn im Frequenzbereich in Hz dann haben Frauen einen größeren Stimmumfang als Männer. Betrachtet man ihn hingegen logarithmisch in Halbtönen, so haben beide Geschlechter einen sehr ähnlichen Stimmumfang [4]. Dieses Verhalten wurde in [5] genauer untersucht. Der Stimmumfang beim Sprechen wird hier durch die durchschnittliche Grundfrequenz und deren dazugehörige Standardabweichung beim Sprechen angegeben. Unter Verwendung der oben angeführten Grundfrequenzen führt das zu einer Standardabweichung von etwa 50 Hz bei Männern und etwa 90 Hz bei Frauen. Jedoch logarithmisch in Halbtönen betrachtet, liegt die Standardabweichung für beide Geschlechter bei $3,4$ Halbtönen.
Eine weitere Eigenschaft von menschlicher Sprache ist die Stimmqualität. Diese steht nicht im direkten Zusammenhang mit der F0, jedoch kann die Stimmqualität prosodische Informationen übermitteln. Die Stimmqualität geht von knarrig bis zu gehaucht. In der Mitte zwischen diesen zwei Extremen liegt die „normale“ oder modale Stimme [6]. Besonders die nicht-modale

Stimmqualität ist in dieser Arbeit von Interesse, weil die Schwingungsmuster von der modalen Schwingung abweichen, was wiederum zu Berechnungsfehlern führen kann. Ursachen für diese Abweichung können Doppelpulse, Perioden mit abwechselnd hoher und niedriger Amplitude, Unregelmäßigkeiten in der Schwingung, oder Maskierung durch Rauschen (in behauchter Stimme), sein [7].
Diese verschiedenen Quellen für Berechnungsfehler können in prosodischen Studien zu Problemen führen, insbesondere wenn immer wieder derselbe Berechnungsfehler auftritt, da es dadurch zu fälschlichen Regelmäßigkeiten kommen kann, die in einer quantitativen Analyse als prosodisches Muster interpretiert werden könnten, obwohl sie nicht die Realität abbilden.

Das Ziel dieser Arbeit ist es, einen möglichst exakten F0-Verlauf zu erzeugen, um Sprachmelodie abzubilden. Es gibt einige vielversprechende Algorithmen zur Berechnung der F0, die genau das versprechen. Jedoch gibt es auch hier Nachteile. Oft wird der F0-Verlauf geglättet, um Berechnungsfehler zu eliminieren [8], was wiederum zur Verfälschung des F0-Verlaufs führt und weitergehend zum möglichen Verlust prosodischer Information; oder es wird nur mit synthetischen Daten gearbeitet [9], was natürlich zu guten Ergebnissen führt, jedoch in der Praxis mit „echten“ Daten ganz andere Ergebnisse liefern kann.
Um sicherzustellen, dass der F0-Verlauf möglichst fehlerfrei ist, wurde in dieser Arbeit ein Tool entwickelt, das eine halb-automatische Korrektur falsch berechneter F0-Werte ermöglicht. Dafür werden die Ursachen sowie die Auswirkungen der Berechnungsfehler genauer betrachtet. Eine halb-automatische Korrektur ist deswegen notwendig, da die automatische Korrektur durch einen Algorithmus wieder zu neuen Artefakten führen kann. Da ein Algorithmus strikt nach einem bestimmten Muster arbeitet, würde er Dinge korrigieren, die zwar vom Algorithmus als falsch detektiert wurden, aber in Wirklichkeit gar nicht falsch sind. Dies würde wiederum zu falschen Interpretationen in der prosodischen Analyse führen und macht einen manuellen Zwischenschritt notwendig. In dieser Arbeit werden außerdem empirisch ermittelte Werte verwendet, die sich im Entwicklungsprozess als sinnvoll herausgestellt haben. Diese können jedoch wieder eine Quelle für fehlerhafte Korrekturen sein, was wiederum den manuellen Evaluierungsschritt motiviert.

2 F0-Berechnung und Datensatz

2.1 F0-Berechnung

Es gibt viele verschiedene Methoden, um die F0 von Sprache zu ermitteln. Bei allen Methoden wird versucht, mögliche Kandidaten für die F0 über die Quasi-Periodizität des Sprachsignals zu ermitteln. Es wird immer mehrere mögliche Kandidaten geben, da nicht nur die Grundschwingung, sondern auch die Oberschwingungen Quasi-Periodizität aufweisen. Jedoch ist die Grundschwingung meist am stärksten ausgeprägt und somit der wahrscheinlichste Kandidat für die F0 [10]. Wegen der Quasi-Periodizität und des schnellen Wechsels der Grundfrequenz und der Formanten beim Sprechen, ist es notwendig, Sprache in kürzeren Abschnitten (Fenstern) zu verarbeiten, da andernfalls die Auflösung darunter leidet. Es gibt verschiedene Ansätze für F0-Berechnungs-Algorithmen (siehe Abschnitt 2.3), die im Frequenz- oder Zeitbereich arbeiten.

2.2 F0-Berechnungsfehler

Algorithmen zur F0-Berechnung sind auffällig für bestimmte Fehler, wobei die zwei häufigsten sind [11],

1. dass eine vorhandene Grundfrequenz falsch erkannt wird; meist um eine Oktave verschoben. In der Phonetik nennt man diesen Fehler „Oktavsprung“. Da diese jedoch nicht immer genau um eine Oktave springen, werden diese Fehler in dieser Arbeit als „Frequenzsprünge“ bezeichnet.

2. dass fälschlicherweise eine Grundfrequenz berechnet wird, obwohl keine vorhanden ist. In dieser Arbeit wird dieser Fehler als „Fehl-F0“ bezeichnet.

Frequenzsprünge treten auf, wenn die Periodizität falsch erkannt worden ist. D.h. es ist genau die doppelte oder halbe Periodendauer erkannt worden und somit auch die halbe oder doppelte Frequenz. Diese Fehler können nur in stimmhaften Lauten auftreten, da bei stimmlosen keine Periodizität vorliegt (siehe Abbildung 2.1). Dargestellt ist hier ein Spektrogramm und die in Praat berechnete F0 (blaue Linie). Der Berechnungsfehler ist in diesem Fall um eine Oktave nach unten im Vokal [i] von dem Wort „Tonstudio“ passiert.

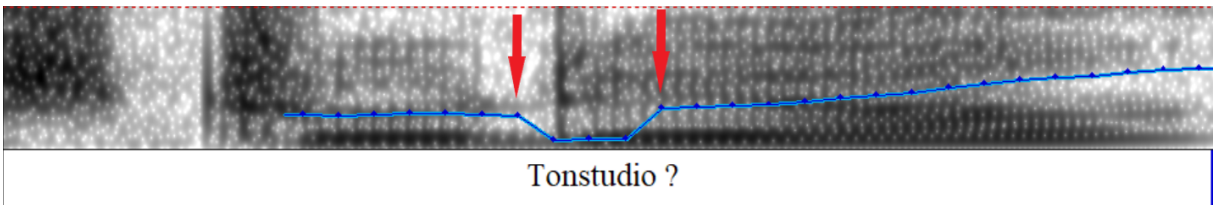


Abbildung 2.1: Ausschnitt aus Praat: Fall eines Oktavsprungs im Übergang von [d] zu [i]

Eine Fehl-F0 tritt in stimmlosen Teilen der Sprache auf und ist dadurch zu erklären, dass in einem Rauschen (z.B. Frikative) eine Regelmäßigkeit erkannt wird. Abbildung 2.2 zeigt einen solchen Fall. Es wurde eine F0 in dem Frikativ [x] von dem Wort „machen“ erkannt.

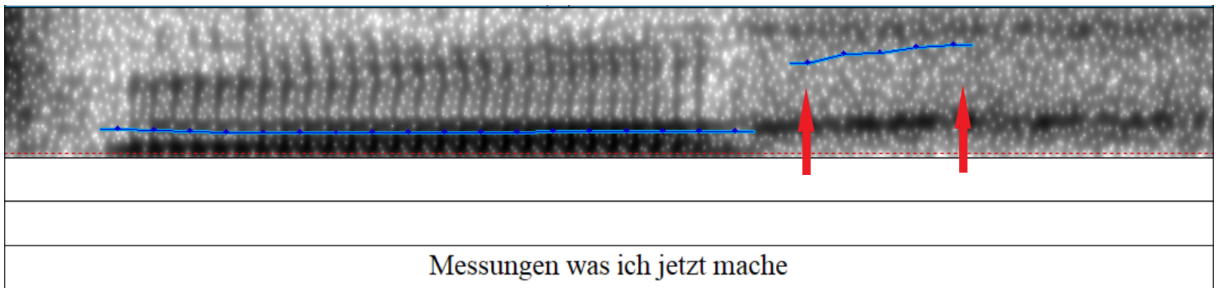


Abbildung 2.2: Ausschnitt aus Praat: Fall einer Fehl-F0 in einem [x]

Um Berechnungsfehler von tatsächlichen F0-Verläufen abzugrenzen, muss geklärt werden, welche F0-Bewegungen für den Menschen überhaupt möglich sind. Ein Mensch kann eine Frequenzbewegung von 12 Halbtönen (eine Oktave) schnellstens in etwa 150 ms vollziehen [12]. Dies würde bei einer Auflösung von 10 $^{ms}/_{Sample}$ (diese wird später für die Analyse verwendet) unter der Annahme von Linearität bedeuten, dass nicht einmal ein Sprung von einem Halbton pro Sample möglich wäre. Mit dieser Information ist es möglich, eine Schwelle zu definieren, die eine klare Abgrenzung zwischen möglichen und unmöglichen Frequenzbewegungen zieht (siehe Unterunterabschnitt 3.3.3).

2.3 F0-Tracking-Algorithmen

Für die vorliegende Arbeit ist der Praat-Algorithmus gewählt worden [13]. Der F0-Verlauf wird hierbei nicht nachträglich durch Glättung verfälscht und er ist ein viel verwendeter Algorithmus. Ebenso ist Praat das Standardprogramm für die Sprachanalyse in der Phonetik, was den Praat-Algorithmus zum perfekten Kandidaten für diese Arbeit macht. Ein weiterer Pluspunkt ist die Benutzerfläche rund um den Algorithmus. Das Programm Praat bietet viele verschiedene Möglichkeiten um Sprachsignale zu analysieren, was für die Auswertung der Ergebnisse von großem Vorteil ist.

2.4 Datensatz

Der verwendete Datensatz ist das GRASS Korpus (Graz corpus of Read And Spontaneous Speech) [14]. GRASS besteht aus Aufnahmen von gelesenem österreichischem Deutsch, sowie Konversationssprache. Insgesamt besteht das Korpus aus etwa 1900 Minuten Sprache von 38 Sprecher:innen. In dieser Arbeit wird ein Teil der Konversationssprache verwendet. Hierbei handelt es sich um spontane Gespräche (ohne jegliche Vorgabe) zwischen Freunden, Partnern, Familienmitgliedern oder Kollegen. Diese sprechen österreichisches Deutsch und wechseln öfter in den Dialekt. Die Gespräche sind je eine Stunde lang und wurden orthographisch annotiert. Diese Annotationen stehen in einer zusätzlichen Textdatei, dem Textgrid (siehe Abbildung 2.3). Aus diesem Korpus wurden für die Analyse des implementierten Algorithmus drei Gesprächsteile zu je 15 Minuten ausgewählt:

- Gespräch zwischen zwei Männern (013M und 014M)

- Gespräch zwischen zwei Frauen (038F und 039F)
- Gespräch zwischen einer Frau und einem Mann (025F und 005M)

Diese sechs Sprecher:innen haben sehr unterschiedliche Stimmen, womit man viele Fälle für die Analyse abdecken kann. Die Aufnahmen sind im Format `.wav` und in Stereo aufgenommen. Hierbei bezieht sich jeweils ein Kanal auf eine:n Sprecher:in. Im Textgrid wurde genau annotiert, was welche:r Sprecher:in zu welchem Zeitpunkt sagt, und in einem eigenen „Tier“ vermerkt (siehe Abbildung 2.3). Die blauen Trennungslinien im Textgrid sind die zeitlichen Grenzen des Gesagten. Diese zeitlichen Intervalle werden im Folgenden als „Chunks“ bezeichnet. Die blaue Kurve zeigt den berechneten F0-Verlauf und oben ist sehr gut zu erkennen, welcher Kanal zu welcher Person gehört.

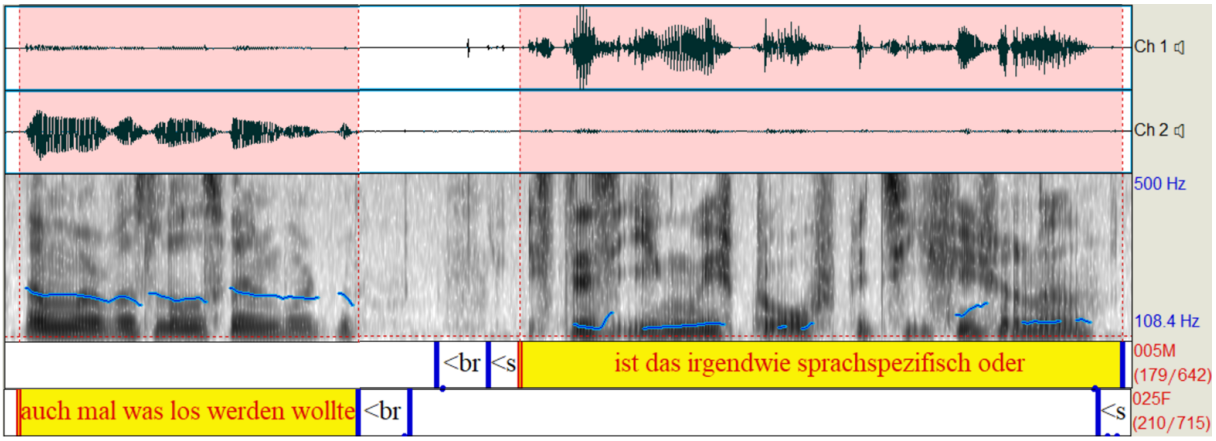


Abbildung 2.3: Ausschnitt eines Textgrids von 005M und 025F

Einige Chunks tragen keinerlei Information über die Grundfrequenz und sind somit vernachlässigbar für diese Untersuchung und können verworfen werden. Das Sprachsignal in diesen Chunks ist geräusch-ähnlich, wie z.B. Atmen, Schnalzen der Zunge oder Rauschen aus der Umgebung oder der sprechenden Person. Ein weiterer Sonderfall ist Lachen. Der Grundfrequenzverlauf von Lachen schwankt besonders stark, weshalb Chunks, die Lachen beinhalten, in dieser Arbeit aussortiert werden.

Beim gleichzeitigen Sprechen beider Personen kann es vorkommen, dass ein:e Sprecher:in auf dem Kanal des:der anderen Sprechenden zu hören ist. Dies kann zur Verfälschung der berechneten F0-Kurve führen und weiters zur falschen Erkennung von Fehlern. Deswegen werden alle Chunks, bei denen dies vorkommt, ebenfalls verworfen.

3

Detektion und Verarbeitung von F0-Berechnungsfehlern

In diesem Kapitel wird die Herangehensweise zur Detektion von F0-Berechnungsfehlern sowie deren Verarbeitung näher betrachtet. Es wird auf das entwickelte Tool und die verwendeten Parameter eingegangen und die manuelle Weiterverarbeitung der Ergebnisse durch den Menschen wird erläutert.

3.1 Implementierung

Die Entwicklung des Tools ist vollständig in `Python` (Version 3.8) [15] unter Verwendung einiger Pakete erfolgt. `Parselmouth` [16] ist ein Paket zur Implementierung des Praat-Algoritmus (siehe Abschnitt 2.3) in Python. Somit ist es besonders gut für die Berechnung des F0-Verlaufs geeignet, da sehr ähnliche Ergebnisse wie durch die direkte Verwendung von Praat erzeugt werden. `TextGridTools` [17] wird zur Verarbeitung der Textgrids verwendet. Alle mathematischen Berechnungen erfolgen mittels `Numpy` [18].

3.2 Algorithmus und allgemeiner Arbeitsablauf

In Abbildung 3.1 ist der grundlegende Arbeitsablauf des Tools dargestellt. Die Eingangsgrößen sind die jeweils zusammengehörigen Audiodateien und Textgrids aus dem in Abschnitt 2.4 beschriebenen Datensatz. Wichtige Parameter, wie der F0-Median (f_{med}) der Sprecher:innen sowie einige Konstanten, werden aus einer Konfigurationsdatei eingelesen. Die Ausgabe des Tools besteht aus den detektierten Fehlern sowie zugehörigen Korrekturvorschlägen. Die Korrekturvorschläge werden manuell evaluiert und nötige Änderungen im Textgrid eingetragen. Diese modifizierten Textgrids werden als neue Eingangsgröße für einen erneuten Durchlauf des Algorithmus verwendet. In diesem erneuten Durchlauf werden sämtliche Korrekturen angewendet; d.h. alle im ersten Durchlauf detektierten Berechnungsfehler inklusive der manuell evaluierten werden ausgebessert. Die endgültige Ausgabe besteht aus sämtlichen im jeweiligen Sprachsignal gemessenen sowie korrigierten F0-Werten mit den zugehörigen Zeitwerten. Zusätzlich werden für alle korrigierten Stellen Grafiken exportiert, aus denen die durch die Korrektur erfolgte Änderung ersichtlich ist.

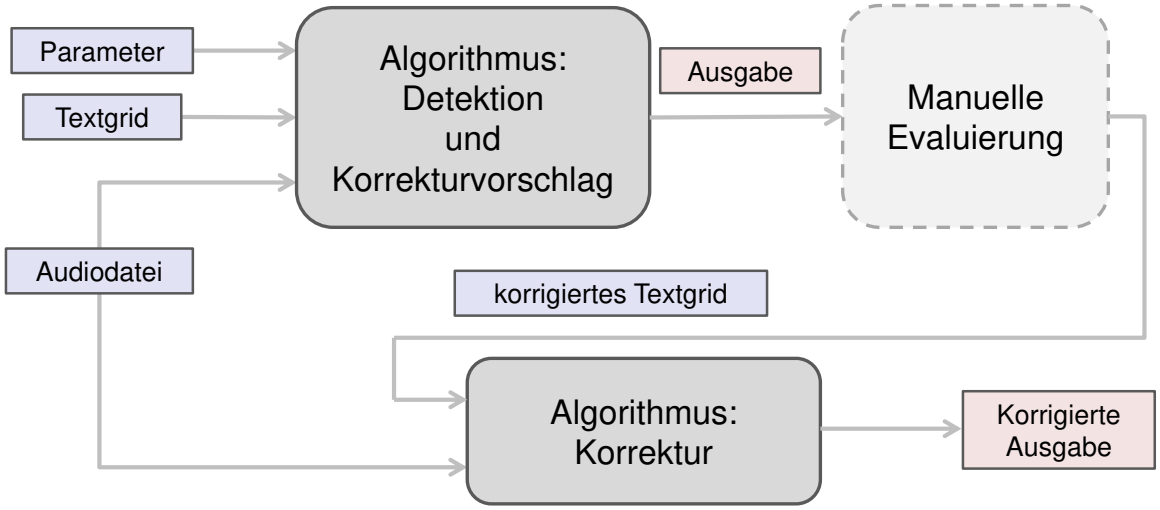


Abbildung 3.1: Flussdiagramm des Arbeitsablaufs

3.3 Detektion von F0-Berechnungsfehlern

Nun wird die genaue Funktionsweise des in Abbildung 3.1 dargestellten Detektions-Algorithmus betrachtet. Eine genauere Darstellung des Algorithmus zeigt Abbildung 3.2. Die einzelnen Blöcke werden in den nächsten Abschnitten genauer erklärt.

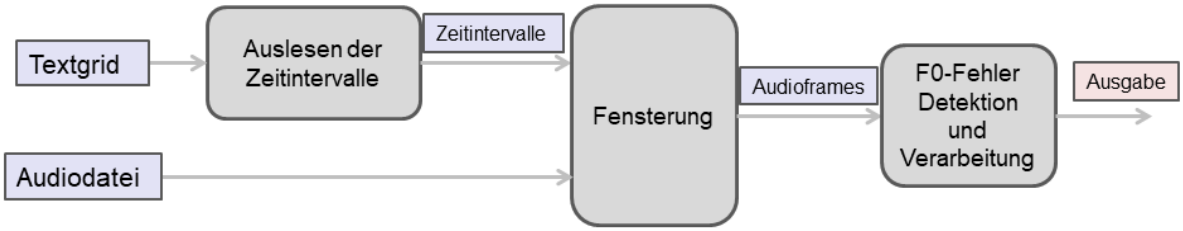


Abbildung 3.2: Schematische Darstellung des Algorithmus: Detektion und Korrekturvorschlag

3.3.1 Auslesen der Zeitintervalle

Da für die Analyse von F0-Berechnungsfehlern nur Bereiche von Interesse sind, in denen tatsächlich gesprochen wird, werden ausschließlich diese Zeitintervalle (Chunks; vgl. Abschnitt 2.4) des Textgrids für die Verarbeitung der Audiodatei verwendet. Um die Zeitintervalle zu extrahieren, werden die jeweiligen Tiers eingelesen und in die einzelnen Chunks aufgeteilt.

3.3.2 Fensterung der Audiodatei

Mit den aus dem Textgrid ausgelesenen Zeitintervallen kann die Audiodatei in kurze Audioframes unterteilt werden. Der Praat-Algorithmus sowie dessen Python-Entsprechung *Parselmouth*

arbeiten in der Zeitdomäne. Daher sind Auflösung und Qualität der Berechnungen stark abhängig von der Länge des zu untersuchenden Signals.

3.3.3 F0-Fehlerdetektion und -verarbeitung

In diesem Teil des Arbeitsablaufs erfolgt die eigentliche Untersuchung der Grundfrequenzverläufe. Jedoch müssen die Audiodaten zunächst aufbereitet werden, bevor die Fehler detektiert und dargestellt werden können. Dieser Vorgang ist in Abbildung 3.3 schematisch dargestellt.

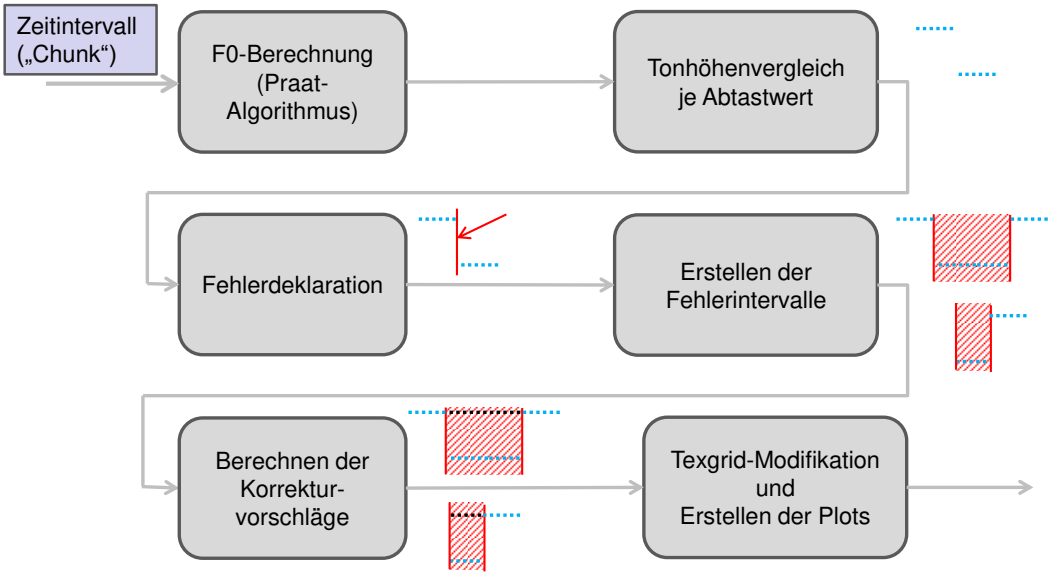


Abbildung 3.3: Flussdiagramm der F0-Fehlerdetektion und Verarbeitung

F0-Berechnung (Praat Algorithmus)

Im ersten Schritt werden die Audioframes analysiert und der Grundfrequenzverlauf extrahiert. Man kann diesen Verlauf als Funktion $f[n]$ darstellen. Die Parameter $f_{0,\max}$ und $f_{0,\min}$ schränken den Bereich ein, in dem eine F0 erwartbar ist. Dadurch werden ausschließlich F0-Werte von $f_{0,\min} \leq f \leq f_{0,\max}$ berechnet. Aus der Information über die durchschnittliche Grundfrequenz der menschlichen Stimme wurden diese Grenzen wie folgt gewählt:

- $f_{0,\min} = 80\text{Hz}$ und $f_{0,\max} = 500\text{Hz}$ bei Frauen
- $f_{0,\max} = 60\text{Hz}$ und $f_{0,\max} = 400\text{Hz}$ bei Männern

Die Auflösung für die Berechnung der F0-Werte beträgt in etwa $10\,{}^{\text{ms}}/_{\text{sample}}$. Abhängig von der Länge des zu untersuchenden Signals schwankt die Auflösung um zirka $\pm 1\%$.

Tonhöhenvergleich je Abtastwert

Im nächsten Schritt wird der F0-Verlauf sample-weise analysiert. Das bedeutet, dass jeweils zwei benachbarte Frequenzwerte miteinander verglichen werden. Um Fehler zu detektieren, wird für jeden Abtastwert $f[n]$ die relative Frequenzänderung f_{rel} zum vorherigen Abtastwert $f[n-1]$ mittels Gleichung 3.1 berechnet. Da für den ersten Abtastwert noch kein vorheriger Wert vorliegt,

wird in diesem Fall der F0-Median des:der Sprecher:in verwendet, da dieser meist einen guten Vergleichswert bietet.

$$f_{\text{rel}} = \frac{\max(f[n], f[n-1])}{\min(f[n], f[n-1])} \tag{3.1}$$

Durch die Verwendung des Minimums und Maximums muss die relative Frequenzänderung immer größer oder gleich 1 sein, was die spätere Weiterverarbeitung vereinfacht, da man nur eine Schwelle anstatt von zwei definieren muss (siehe Unterunterabschnitt 3.3.3). Jedoch geht Information über die Richtung, d.h. Frequenzerhöhung oder Senkung, verloren. Diese wird folgendermaßen wieder hergestellt:

$$\text{Richtung der Frequenzänderung} = \begin{cases} 1 & \text{für} \quad f[n] > f[n-1] \\ -1 & \text{für} \quad f[n] < f[n-1] \end{cases} \tag{3.2}$$

wobei 1 einer Frequenzerhöhung und -1 einer Frequenzsenkung vom vorigen zum aktuellen Abtastwert entspricht. Um zu bestimmen, ob eine Frequenzänderung ein Berechnungsfehler, oder tatsächlich im Signal vorhanden ist, wird der Wert der Frequenzänderung mit der maximal menschlich möglichen Frequenzbewegung von etwa 12 Halbtönen innerhalb von 150 ms verglichen (siehe Abschnitt 2.2). Frequenzänderungen von mehr als einem Halbton treten jedoch regelmäßig im Sprachsignal auf. Um diese nicht als Berechnungsfehler zu erfassen, wird eine untere Fehlerschwelle $f_{\text{rel,min}}$ eingeführt. Diese ist abhängig von dem Parameter Δf_{tol}, der den Toleranzbereich für zulässige Frequenzänderungen beschreibt, und hat den Zusammenhang:

$$f_{\text{rel,min}} = 2 - \Delta f_{\text{tol}} \qquad f_{\text{rel,max}} = 2 + \Delta f_{\text{tol}} \tag{3.3}$$

Hierbei steht die Zahl 2 für die relative Frequenzänderung bei einem Sprung von einer Oktave beziehungsweise 12 Halbtönen. Dadurch werden kleinere tatsächliche Schwankungen der F0 nicht als Berechnungsfehler detektiert. Der Parameter Δf_{tol} wurde empirisch ermittelt und hat für die in dieser Arbeit verwendeten Daten den Wert $\Delta f_{\text{tol}} = 0,3$. Dies führt zu einer unteren Grenze von $f_{\text{rel,min}} = 1,7$ und einer oberen Grenze von $f_{\text{rel,max}} = 2,3$. Für einen detektierten Wert $f[n] = 114\,\text{Hz}$ würde das bedeuten, dass die Frequenzänderung zur nächsten Abtastposition $f[n+1]$ als Berechnungsfehler detektiert wird, wenn gilt: $f[n+1] < 67\,\text{Hz}$ oder $f[n+1] > 194\,\text{Hz}$.

Die detektierten Frequenzänderungen und die Abtastposition n werden für die Deklaration zwischengespeichert. Für die spätere Weiterverarbeitung durch den Menschen wird die relative Frequenzänderung mittels der Gleichung 3.4 in Halbtöne umgerechnet.

$$f_{\text{rel,Halbtöne}} = 12 \cdot \log_2(f_{\text{rel}}) \tag{3.4}$$

Mit dem Wert für $\Delta f_{\text{tol}} = 0,3$ kommt man so auf einen Bereich von etwa 9 bis 14,5 Halbtönen.

Fehlerdeklaration

Die detektierten Frequenzänderungen werden im nächsten Schritt ihrer Fehlerart zugeordnet. Wie in Abschnitt 2.2 beschrieben, wird auf zwei Fehlerarten eingegangen: Frequenzsprünge (`jump`), und Fehl-F0 (`fake`). Falls die Frequenzänderung im Bereich um 12 Halbtöne liegt, wird angenommen, dass es sich um einen Kandidaten für einen Frequenzsprung (`jump`) handelt. Bei einer Frequenzänderung, die größer ist als die obere Schwelle, wird dies ein Kandidat für eine

Fehl-F0 (`fake`). Dieser Zusammenhang ist in Gleichung 3.5 dargestellt.

$$\text{Kandidat für die Art des Fehlers} = \begin{cases} \texttt{jump} & \text{für} \quad f_{\text{rel,min}} < f_{\text{rel}} < f_{\text{rel,max}} \\ \texttt{fake} & \text{für} \quad f_{\text{rel}} > f_{\text{rel,max}} \end{cases} \tag{3.5}$$

Es kann jedoch vorkommen, dass einzelne Abtastwerte stärker oder schwächer abweichen als der Rest der Werte innerhalb eines Fehlerintervalls. In einem solchen Fall wird überprüft, in welchem Frequenzbereich sich diese Abtastwerte befinden. Ist z.B. eine Fehl-F0 detektiert worden, aber die Frequenz der entsprechenden Abtastwerte liegt in einem für Frequenzsprünge typischen Bereich, wird angenommen, dass es sich hierbei um einen Frequenzsprung mit untypisch hoher Sprunggröße handelt ($f_{\text{rel}} > f_{\text{rel,max}}$). Wenn hingegen ein Frequenzsprung erkannt wird, die Frequenz der Abtastwerte aber in einem untypischen Bereich für Frequenzsprünge liegt, wird wiederum angenommen, dass es sich hier um eine Fehl-F0 mit untypisch niedriger Sprunggröße handelt ($f_{\text{rel}} < f_{\text{rel,max}}$). Der für Frequenzsprünge typische Frequenzbereich ist jener Bereich, in dem die Frequenz der entsprechenden Abtastwerte kleiner $f_{\text{med}} \cdot \Delta f_{\text{tol}}$ ist; d.h, dass sich der für Frequenzsprünge typische Bereich bis zum $2,3$-fachen des F0-Medians erstreckt. Dies wurde so gewählt, da eine Verdoppelung des Medians als relativ wahrscheinlich erscheint.
Nach der Analyse des F0-Verlaufs werden für detektierte Berechnungsfehler die Abtastposition, die Art (`jump` oder `fake`) und die Größe der fehlerhaften Frequenzänderung (in Halbtönen, sowie absolut, von nun an als „Sprunggröße“ bezeichnet) für die Weiterverarbeitung gespeichert.

Erstellen der Fehlerintervalle

Da sich ein Berechnungsfehler meist über mehrere Abtastwerte erstreckt, sind die detektierten Abtastpositionen allein nicht sehr aussagekräftig. Sie zeigen lediglich Anfang oder Ende eines möglichen Berechnungsfehlers an. Daher wird im nächsten Schritt ein Fehlerintervall auf der Zeitachse erzeugt. Für die spätere Zuordnung der Fehlerintervalle im Korrekturschritt, werden die Intervalle fortlaufend nummeriert. Ein Fehlerintervall schließt alle falsch berechneten Werte ein. Es gibt zwei Möglichkeiten:

Fall 1: Zwei detektierte Fehler gehören zusammen

Fall 2: Der Fehler steht alleine

In Fall 1 wird das Fehlerintervall sofort erstellt. Die im vorherigen Schritt detektierten Fehlerpositionen beschreiben Anfang und Ende des Fehlerintervalls. Da zwei Fehler innerhalb eines Chunks nicht notwendigerweise zusammengehören, müssen sie zusätzliche Kriterien erfüllen, um als ein zusammengehöriges Fehlerintervall deklariert zu werden:

1a Beide Fehler müssen von gleicher Art sein.

1b Die Richtung der Frequenzänderung muss entgegengesetzt sein. z.B zuerst Frequenzänderung nach oben, dann nach unten.

1c Die Fehler dürfen zeitlich nicht zu weit auseinander liegen.

Dies ist in Abbildung 3.4 verdeutlicht, wo ein Frequenzsprung zu sehen ist. Die erste Fehlerposition zeigt einen Sprung nach unten, die zweite einen Sprung nach oben. Durch Beobachtung hat sich herausgestellt, dass sich ein Fehler selten über einen längeren Zeitraum als $0,5$s erstreckt.

Herausfordernder ist das Erzeugen eines Fehlerintervalls in Fall 2. Mögliche Beispiele sind in Abbildung 3.5 dargestellt. Hier ist lediglich die Information über Anfang oder Ende des Fehlers vorhanden. Es kann zwar vorkommen, dass nur ein einzelner Abtastwert fehlerhaft ist, jedoch ist dieser Fall sehr selten und der Anfang würde dem Ende entsprechen. Um die Länge des Fehlerintervalls zu bestimmen, muss als erstes überprüft werden, welche der Frequenzen (d.h.

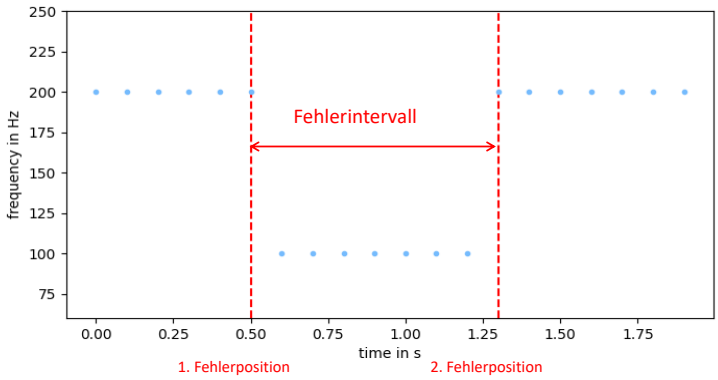
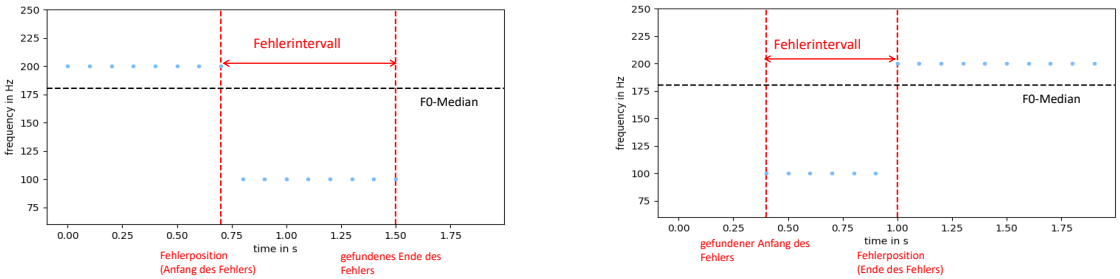


Abbildung 3.4: Schematische Darstellung eines Fehlerintervalls aus zusammengehörigen Fehlerpositionen

die Frequenz vor oder nach dem Sprung) die falsch berechnete ist. Hierzu werden die beiden Frequenzwerte, die unmittelbar vor und nach der Fehlerposition liegen, mit dem F0-Median des:der Sprecher:in verglichen. Diejenige Frequenz, die relativ gesehen näher am Median liegt, gilt als die korrekt berechnete.

Um herauszufinden, ob der Fehler Anfang oder Ende des Fehlerintervalls ist, wird wieder ein Vergleich der unmittelbar vor und nach der Fehlerposition liegenden Frequenzwerte durchgeführt. Wenn die Frequenz *vor* der Fehlerposition als richtig deklariert wurde, gilt die Fehlerposition als Anfang und die falsch berechnete Frequenz befindet sich in der positiven Richtung der Zeitachse (siehe Abbildung 3.5(a)). Wenn hingegen die Frequenz *nach* der Fehlerposition als richtig deklariert wurde, ist die Fehlerposition das Ende des Fehlerintervalls und die falsch berechnete Frequenz befindet sich in der entgegengesetzten (negativen) Richtung der Zeitachse (siehe Abbildung 3.5(b)). Um die Länge des Fehlerintervalls zu bestimmen, wird solange in die Richtung der falsch berechneten Frequenz geprüft, wie die folgenden Frequenzwerte im Bereich der als falsch berechneten Frequenz liegen ($\pm 20\%$). Sobald die Frequenzwerte stärker von der falsch berechneten Frequenz abweichen, wird diese Position als Anfang bzw. Ende angenommen.



(a) Fehlerposition ist der Anfang des Fehlerintervalls (b) Fehlerposition ist das Ende des Fehlerintervalls

Abbildung 3.5: Schematische Darstellung von Fehlerintervallen, die nicht aus zwei zusammengehörigen Fehlerpositionen bestehen.

Falls Fall 1 eingetreten ist, enthält das Fehlerintervall zwei Werte für die Sprunggröße, bei Fall 2 nur einen.

Korrekturvorschläge

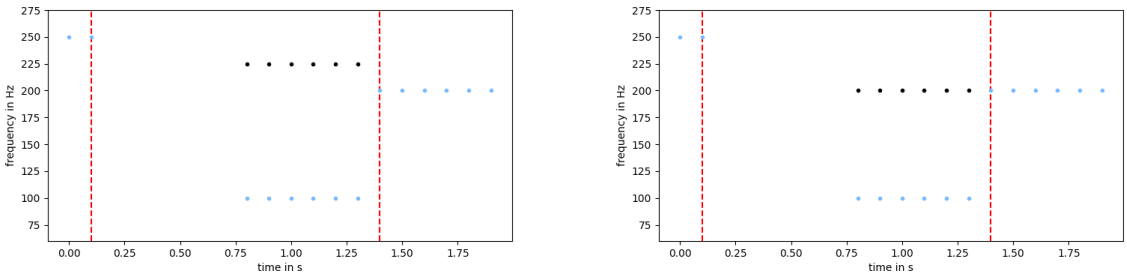
Mit den im vorherigen Schritt erstellten Fehlerintervallen, werden die Korrekturvorschläge der als falsch deklarierten Frequenzwerte berechnet. Die vorgeschlagene Korrektur hängt von der Art des Fehlers sowie der Sprunggröße ab. Im Falle einer Fehl-F0 sollen die betroffenen Frequenzwerte gelöscht werden. Bei den Frequenzsprüngen wird der Korrekturvorschlag f_{corr} für die fehlerhafte Frequenz f_{err} wie folgt berechnet:

$$f_{\text{corr}} = f_{\text{err}} \cdot k \tag{3.6}$$

mit dem Korrekturfaktor

$$k = \begin{cases} f_{\text{rel}} & \text{für Korrektur nach oben} \\ 1/f_{\text{rel}} & \text{für Korrektur nach unten} \end{cases} . \tag{3.7}$$

Der Korrekturfaktor k hängt von Wert und Richtung der Sprunggröße ab. Falls das Fehlerintervall aus zwei Fehlern besteht, sind zwei Werte für f_{rel} vorhanden. Aus Beobachtung war zu erkennen, dass die Mittelung beider Werte zu nicht ausreichend genauen Korrekturvorschlägen führen kann. Dieser Fall tritt oft dann auf, wenn nach einem detektierten Frequenzsprung mehrere Abtastwerte folgen, die keine F0 haben. Dadurch können sich die beiden Sprunggrößen stark unterscheiden. Falls dies zutrifft, wird die Sprunggröße abhängig von der Anzahl der Abtastwerte ohne F0 bestimmt. Ein solcher Fall ist in Abbildung 3.6 zu sehen. In blau sind die berechneten F0-Werte gekennzeichnet, in schwarz der Korrekturvorschlag. Hier ist klar zu erkennen, dass bei dem ersten detektierten Frequenzsprung einige Abtastwerte ohne F0 folgen, bei dem zweiten schließen gleich Abtastwerte ohne Lücke an. Ebenso unterscheiden sich die Sprunggrößen voneinander. Abbildung 3.6(a) zeigt, welcher Fehler durch die Mittelung beider Sprunggrößen für den Korrekturvorschlag entstehen würde. In einem solchen Fall wird die Sprunggröße des zweiten detektierten Frequenzsprunges verwendet (siehe Abbildung 3.6(b)). Würden beiden detektierten Frequenzsprüngen keine oder ähnlich viele (± 1) Abtastwerte ohne F0 folgen, so würde die Sprunggröße wieder aus beiden Werten gemittelt werden.



(a) Korrekturvorschlag; Mittelung beider Sprunggrößen (b) Korrekturvorschlag; Anpassung der Sprunggröße

Abbildung 3.6: Schematische Darstellung eines Fehlerintervalls mit Abtastwerten ohne F0. Vergleich des Korrekturvorschlags zwischen Mittelung der Sprunggrößen und Anpassung der Sprunggröße an direkt anschließende Werte auf einer Seite des Fehlerintervalls

Ausgabe (Plots und Textgridmodifikation)

Für die manuelle Evaluierung ist eine anschauliche Ausgabe der Korrekturvorschläge hilfreich. Es werden Plots erzeugt, in denen der berechnete F0-Verlauf, die Fehlerpositionen und die Korrekturvorschläge beinhaltet sind. Eine solche Ausgabe ist in Abbildung 3.7 zu sehen. Es ist ein Chunk mit zwei Fehlern (Fehl-F0 und Frequenzsprung) zu sehen. In blau ist der ursprünglich

berechnete F0-Verlauf markiert, in rot die Punkte, die gelöscht werden sollen (d.h. Korrekturvorschlag für erkannte Fehl-F0) und in schwarz ist der Korrekturvorschlag des Frequenzsprunges zu erkennen. Die zeitlichen Grenzen der Fehlerintervalle sind als vertikale Linien visualisiert.

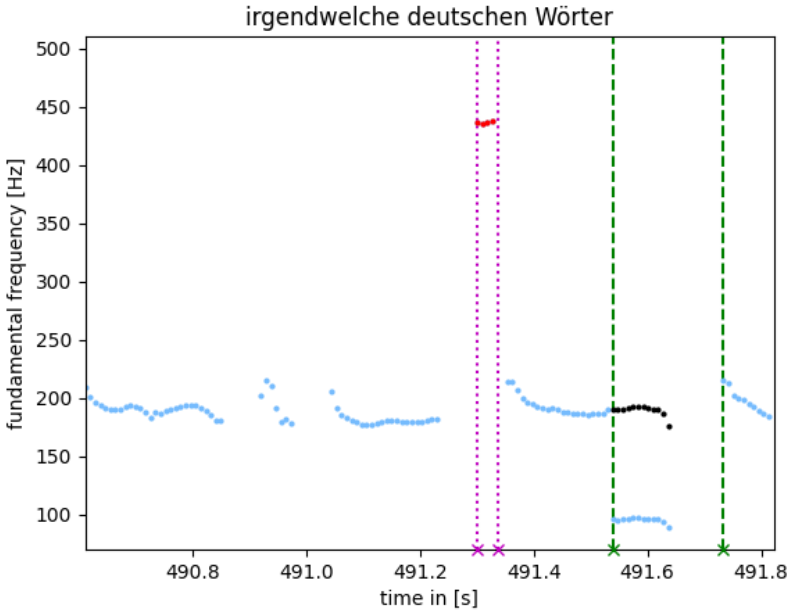


Abbildung 3.7: Berechneter F0-Verlauf in hellblau, Korrekturvorschlag Fehl-F0 in rot, Korrekturvorschlag Frequenzsprung in schwarz, der Titel enthält die gesprochenen Wörter

Um beurteilen zu können, ob ein Korrekturvorschlag richtig oder flasch ist, wird eine präzise zeitliche Zuordnung des Gesprochenen benötigt. Hierfür wird im Textgrid ein neues Tier namens `F0-Fehler` erstellt, welches alle Fehlerintervalle mit der jeweiligen Art des Fehlers und dem entsprechenden Korrekturvorschlag beinhaltet. Die detektierten Fehlerintervalle werden fortlaufend nummeriert. Abbildung 3.8 zeigt ein derart modifiziertes Textgrid. Es sind zwei detektierte Fehlerintervalle eingetragen. Links mit der Intervallnummer 28 ist eine Fehl-F0 (`fake`) zu erkennen. Rechts mit der Intervallnummer 29 ist ein Frequenzsprung (`jump`) mit den Sprunggrößen $-11,9$ und $11,9$ Halbtönen zu sehen.

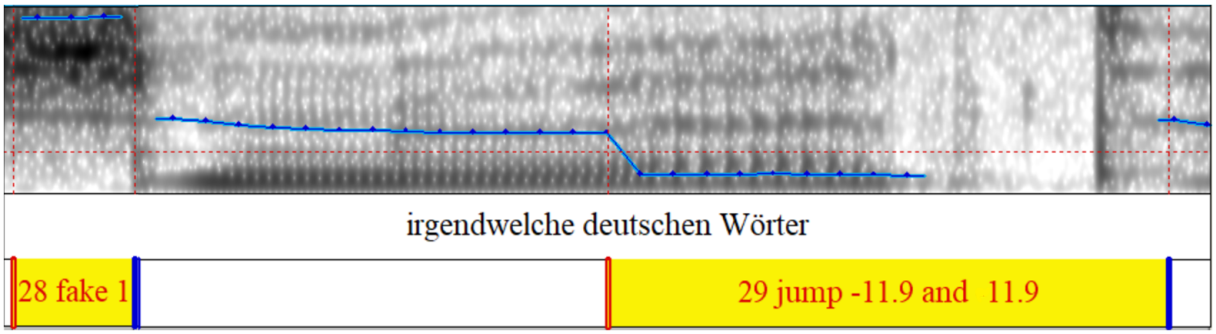


Abbildung 3.8: Ausgabe: Textgridmodifikation. Eine Fehl-F0 gefolgt von einem Frequenzsprung, Sprecherin 038F.

3.4 Manuelle Evaluierung

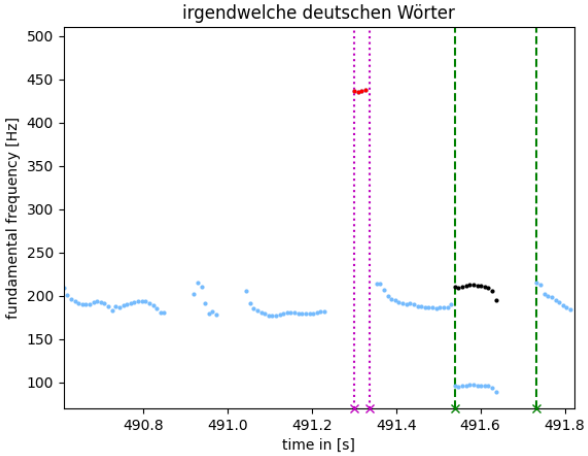
Bei der manuellen Evaluierung werden die Korrekturvorschläge analysiert und ggf. mit der Ausgabe von Praat verglichen. Hierbei sind fünf verschiedene Fälle zu unterscheiden.

1. Ein Fehlerintervall wurde richtig erkannt und der Korrekturvorschlag ist korrekt.
2. Fehldetektion: Ein Fehlerintervall ist falsch detektiert worden d.h. es ist ein Fehlerintervall angezeigt worden, obwohl kein Fehler im betreffenden Bereich vorliegt. Das Fehlerintervall muss gelöscht werden.
3. Die Art des Fehlerintervalls ist falsch erkannt worden und muss korrigiert werden. z.B. wurde eine Fehl-F0 erkannt, wo tatsächlich ein Frequenzsprung vorliegt.
4. Die Sprunggröße ist falsch berechnet worden und muss korrigiert werden.
5. Ein Fehler wurde nicht erkannt und ein neues Intervall wird manuell hinzugefügt.

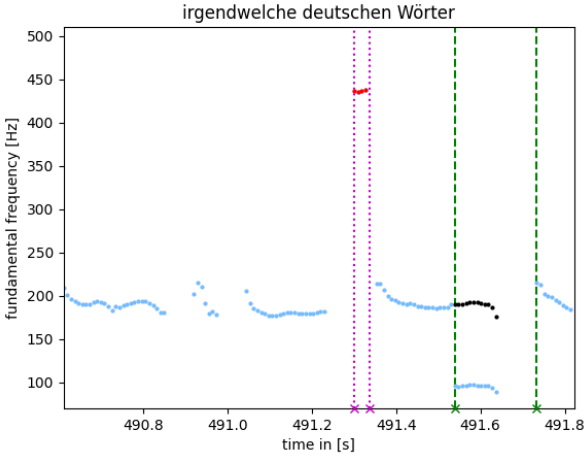
Tritt Fall 1 ein, so müssen keine Änderungen durchgeführt werden. Die Korrekturvorschläge im Ausgabe-Plot können ohne Änderung übernommen werden. Falls einer der Fälle von 2–5 eintritt, wird dies durch die Evaluierung im Textgrid im Tier `F0-Fehler` vermerkt. Im Falle von 2 wird die Information des Fehler-Intervalls gelöscht, bei 3 wird die korrigierte Art angegeben, bei 4 wird der korrigierte Wert angegeben und bei 5 wird ein neues Intervall mit einer neuen Intervallnummer und Art erstellt.

3.5 Korrektur von F0-Berechnungsfehlern

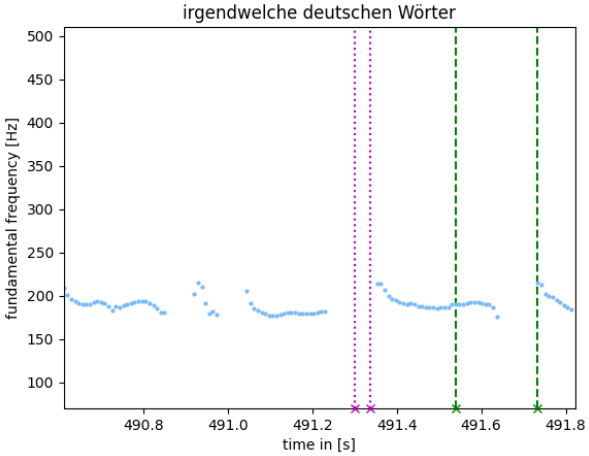
Im letzten Schritt des Ablaufs wird noch einmal derselbe Algorithmus wie in Abschnitt 3.3 verwendet. Die zuvor evaluierten Korrekturvorschläge werden nun auf die Datenpunkte angewendet. Hierzu wird das manuell evaluierte Textgrid eingelesen und anhand der Intervallnummern den detektierten Fehlern zugeordnet. Die manuell evaluierten Fehlerintervalle werden zur eigentlichen Korrektur der F0-Werte verwendet. Abbildung 3.9 zeigt den Ablauf von der Ausgabe mit den Korrekturvorschlägen (3.9(a)), über die Evaluierung (3.9(b)), bis zur korrigierten Ausgabe (3.9(c)). In diesem Beispiel wurde die Sprunggröße des Frequenzsprungs bei der manuellen Evaluierung verändert. Die Art der beiden Fehler wurde nicht verändert, ebenso wurde kein neues Fehlerintervall eingefügt.



(a) Korrekturvorschläge nach Fehlerdetektion



(b) Korrekturvorschläge nach manueller Evaluierung



(c) F0-Verlauf nach Korrektur

Abbildung 3.9: Darstellung des Arbeitsablaufs von Ausgabe-Plot bis zum korrigierten F0-Verlauf am Beispiel einer korrigierten Sprunggröße (Frequenzwerte im grünen Intervall bei ca. 491,6 s)

4 Ergebnisse

In diesem Kapitel werden die Ergebnisse des implementierten Tools näher betrachtet. Die detektierten Berechnungsfehler je Sprecher:in werden genauer auf die Anzahl und Art untersucht. Auf die Zeitspannen der verschiedenen Chunks wird eingegangen, sowie auf die zeitliche Häufigkeit der detektierten Berechnungsfehler. Ebenfalls wird der manuelle Evaluierungsschritt analysiert und auf die Anzahl und Art der manuellen Evaluierungen eingegangen.

4.1 Automatische Berechnungsfehlerdetektion

Tabelle 4.1 zeigt die Anzahl der detektierten Berechnungsfehler nach Sprecher:innen. Da jedoch die Dauer der untersuchten Chunks stark unterschiedlich ist (siehe Abbildung 4.1), sind die absoluten Werte nicht immer aussagekräftig. Bei der Analyse längerer zeitlicher Abschnitte ist es wahrscheinlicher, mehr Fehler zu detektieren als bei kürzeren Abschnitten. Deswegen sind die Ergebnisse noch einmal relativ zur Zeit der Chunks gesehen in Abbildung 4.2 zu sehen. Es sind die Fehler pro Sekunde für jeden Chunk zu erkennen.

ID	# analysierter Chunks	#Chunks mit Fehlern	#Fehler	#Sprünge	#Fehl-F0
025F	267	64 (24%)	129	90	39
038F	61	25 (41%)	34	31	3
039F	166	42 (25%)	67	60	7
005M	123	13 (11%)	14	4	10
013M	115	27 (23%)	38	33	5
014M	323	28 (9%)	34	24	10

Tabelle 4.1: Anzahl detektierte Fehler pro Sprecher:in, aufgeschlüsselt nach Fehlerart

Insgesamt wurden in den drei jeweils 15-minütigen Gesprächen (6 Sprecher:innen) 316 Berechnungsfehler detektiert. Es wurden deutlich mehr Sprünge (242) als Fehl-F0 (76) detektiert. Ebenso ist auffällig, dass bei Frauen mehr als knapp 25% der untersuchten Chunks fehlerbehaftet sind. Bei Männern hingegen sind es durchschnittlich weniger. Bei 014M weisen sogar nur knapp unter 10% der analysierten Chunks Fehler auf. Die Fehler pro Sekunde für jeden Chunk zeigen ein ähnliches Bild. Bei Frauen sind durchschnittlich mehr Fehler pro Sekunde detektiert worden. Die Ausnahme ist 013M. Dieser Sprecher weist sogar den größten Wert für die Fehler pro Sekunde auf. Dies könnte auf die kürzere Dauer der analysierten Chunks zurückzuführen sein. Somit ist klar erkennbar, dass bei Frauen tendenziell mehr Fehler gefunden wurden.
Diese erhöhte Berechnungsfehleranzahl bei Frauen wird besonders bei den Frequenzsprüngen deutlich. Etwa 74.8% (181 von 242) der Frequenzsprünge traten bei Frauen auf. Dies ist durch die durchschnittlich höhere Grundfrequenz zu erklären. Aus Beobachtungen wurde ersichtlich, dass Frequenzsprünge häufiger nach unten auftreten. Unter der Annahme, dass eine Frau eine durchschnittliche F0 von 188 Hz hat, was dem durchschnittlichen F0-Median der drei Sprecherinnen entspricht, würde ein Oktavsprung nach unten 94 Hz ergeben, was im möglichen Bereich

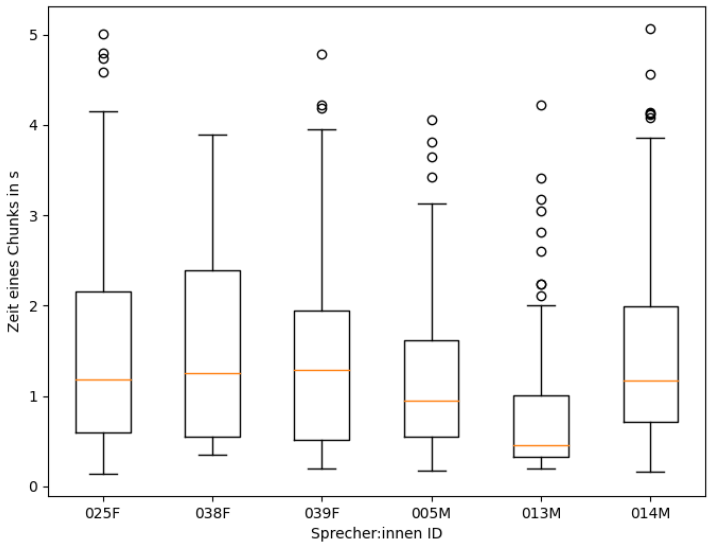


Abbildung 4.1: Boxplot der Chunkdauern, aufgeschlüsselt nach Spercher:innen ID

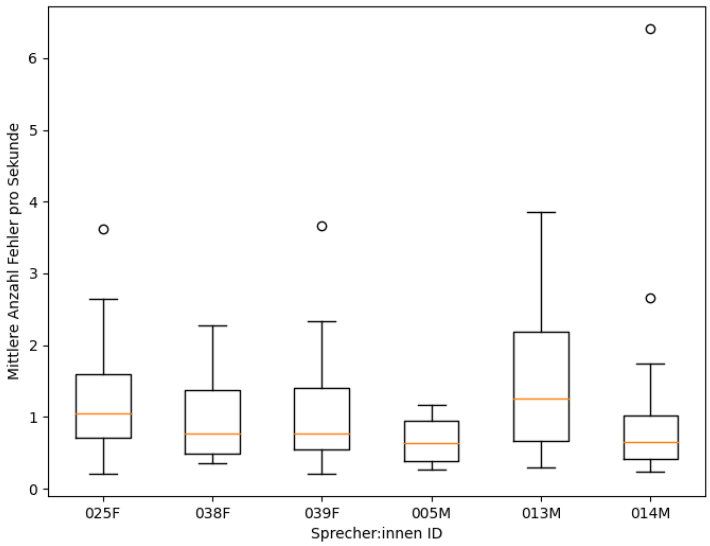


Abbildung 4.2: Boxplot der Fehler pro Sekunde eines Chunks, aufgeschlüsselt nach Sprecher:innen ID

von Sprache, sowie auch im gewählten F0-Bereich des Algorithmus liegt. Hingegen bei einem Mann mit einer durchschnittlichen F0 von 118 Hz (durchschnittlicher F0-Median der drei Sprecher) würde ein Oktavsprung nach unten zu einer Frequenz von 59 Hz führen, was unter dem gewählten F0-Bereich des Algorithmus liegt. So lässt sich die deutlich höhere Anzahl an Sprüngen bei Frauen als bei Männern erklären. Die Häufigkeit der Fehl-F0 ist gleichmäßig über die Geschlechter verteilt. Dies ist nicht überraschend, weil das Fehler sind, die unabhängig von der eigentlichen Grundfrequenz des/der Sprecher:in sind und sich aus einer gewissen Periodizität in geräuschhaften Lauten ergibt (vgl. Abschnitt 2.2).
Auffällig ist, dass Frequenzsprünge häufig in Bereichen von veränderter Stimmqualität auftreten. Speziell bei knarrig gesprochenen Vokalen kam es vermehrt zu Frequenzsprüngen.

4.2 Manuelle Evaluierung

In Tabelle 4.2 sind die Ergebnisse der manuellen Evaluierung zusammengefasst. Wie in der Tabelle zu sehen ist, sind keine Werte für die Sprecherin 039F eingetragen. Die Sprecherin ist eine ältere Dame und ihre Stimme ist generell knarrig. Bei derart veränderter Stimmqualität haben Algorithmen zur F0-Berechnung prinzipiell große Probleme. Durch die nicht-modale Stimmqualität kann es zu sehr extremen F0-Verläufen kommen, die nicht eindeutig vom Menschen als Frequenzsprung oder anderweitige Berechnungsfehler erkennbar sind. Ebenso sind auch viele tatsächliche F0-Verläufe wegen starker Unstetigkeiten fälschlicherweise als Berechnungsfehler detektiert worden. Ein solcher Fall ist in Abbildung 4.3 zu sehen. Besonders in der ersten Sekunde des Chunks schwankt die Grundfrequenz sehr stark. Jedoch sind diese Schwankungen nicht immer konkrete Sprünge, sondern können auch sehr steile, aber kontinuierliche Anstiege oder Abstiege sein. Im Vergleich dazu schwankt bei Sprecherin 038F (Abbildung 3.7) die Grundfrequenz nur sehr wenig. Lediglich bei den Fehlerpositionen ist die Schwankung sehr stark, aber auch sprunghaft. Aus diesem Grund ist Sprecherin 039F aus der Evaluierung ausgenommen worden.

ID	Korrekturen an Fehler-Intervallen	Neue Fehlerintervalle
025F	7% (9/129)	14
038F	0% (0/34)	2
039F	-	-
005M	42,9% (6/14)	2
013M	5,7% (2/35)	5
014M	5,9% (2/34)	2

Tabelle 4.2: Übersicht über die durchgeführten Korrekturen im manuellen Evaluierungsschritt

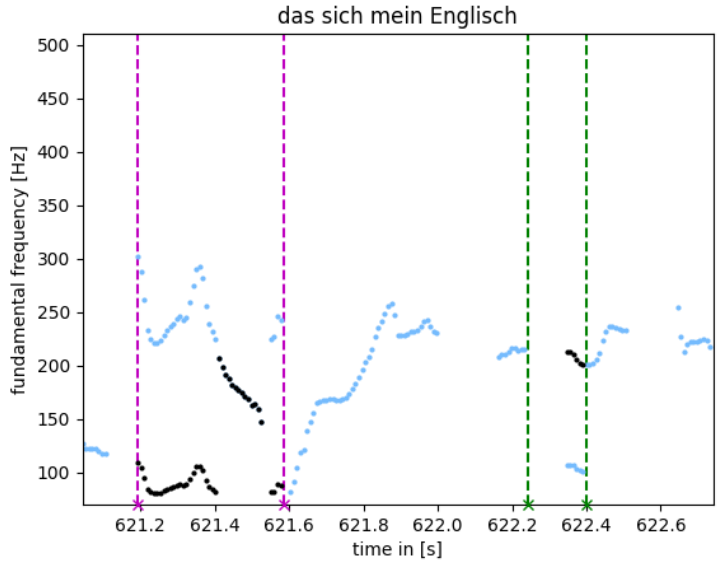
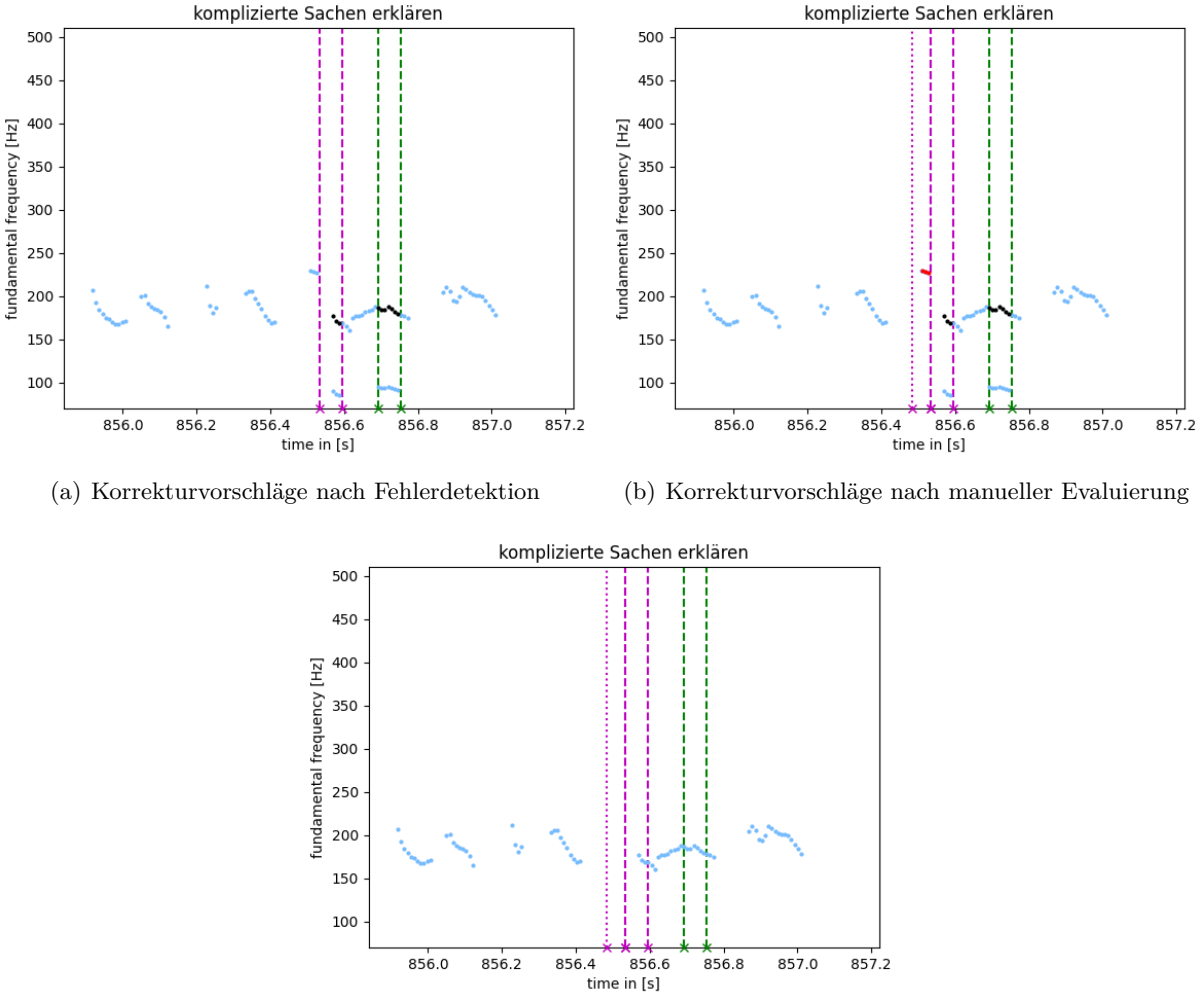


Abbildung 4.3: Beispiel eines Ausgabe-Plots von 039F mit extremem F0-Verlauf und falschem Korrekturvorschlag

Im manuellen Evaluierungsschritt wurden an etwa 10% (25 von 249) der detektierten Fehlerintervalle Korrekturen durchgeführt. Viele dieser Korrekturen beziehen sich auf die Art (12) und die Sprunggröße (10). Nur 3 Fehler-Intervalle sind gänzlich falsch detektiert worden und mussten gelöscht werden. Ebenso sind mehrere neue Fehler-Intervalle (25) eingefügt worden, die zumeist für die Trennung aufeinander folgender Fehler verwendet worden sind. D.h., dass bei zeitlich sehr dicht beieinander liegenden Berechnungsfehlern oft einer der beiden Berechnungsfehler nicht als ein solcher erkannt wird. Mit der Einführung eines neuen Intervalls kann dieses Problem behoben werden. Eine nicht erkannte Fehl-F0 direkt gefolgt von einem Frequenzsprung ist in Abbildung 4.4 zu sehen. In 4.4(b) ist das neu eingefügte Intervall als Fehl-F0 direkt vor dem Frequenzsprung zu erkennen.



(a) Korrekturvorschläge nach Fehlerdetektion

(b) Korrekturvorschläge nach manueller Evaluierung

(c) F0-Verlauf nach Korrektur

Abbildung 4.4: Darstellung des Arbeitsablaufs von Ausgabe-Plot bis zum korrigierten F0-Verlauf am Beispiel eines hinzugefügten Intervalls (Fehl-F0 vor Frequenzsprung)

Abbildung 4.5 zeigt eine Zusammenfassung der oben gezeigten Ergebnisse. Die Frequenzsprünge und Fehl-F0s mit manueller Korrektur beziehen sich auf Frequenzsprünge und Fehl-F0s bei denen die Art oder Sprunggröße verändert wurde. D.h. ein Frequenzsprung bei dem im manuellen Korrekturschritt die Art zu einer Fehl-F0 geändert wurde, scheint hier trotzdem noch als Frequenzsprung mit manueller Korrektur auf (gleiches gilt für die Fehl-F0 mit manueller Korrektur). Hier ist noch einmal der Unterschied zwischen den Geschlechtern zu erkennen. Ebenso ist gut zu erkennen, dass im Verhältnis nur sehr wenig im manuellen Evaluierungsschritt ausge-

bessert werden musste. Lediglich das Einfügen von neuen Intervallen war vermehrt notwendig.

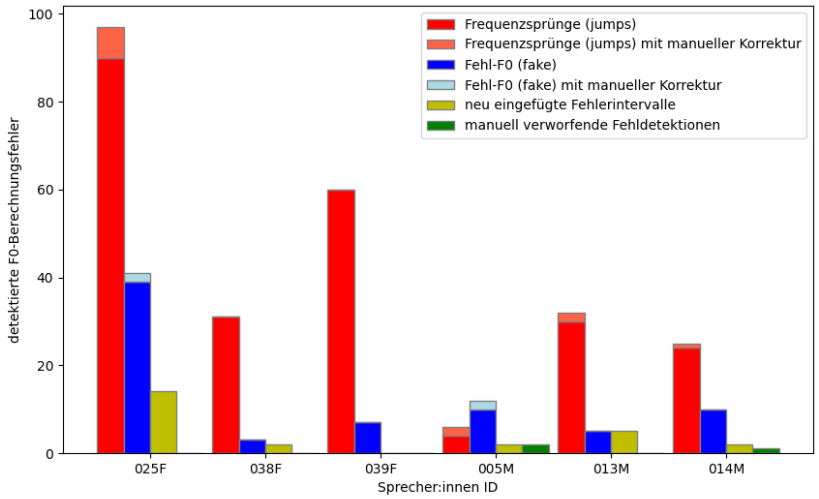


Abbildung 4.5: Zusammenfassung der Ergebnisse der Berechnungsfehlerdetektion sowie der manuellen Evaluierung, aufgeschlüsselt nach Sprecher:innen ID

5 Zusammenfassung und Ausblick

Ziel dieser Arbeit war es, einen möglichst exakten Verlauf der Grundfrequenz (F0) gesprochener Sprache darzustellen. Hierzu wurde ein Tool entwickelt, das eine halb-automatische Korrektur eines zuvor berechneten F0-Verlaufs ermöglicht.
In der Phonetik ist eine korrekt berechnete F0 sehr wichtig, um die Sprachmelodie in Untersuchungen zu Prosodie gut darstellen zu können. Jedoch treten bei der F0-Berechnung häufig Fehler auf, beispielsweise bei nicht-modaler Stimmqualität oder wenn Periodizität in geräuschhaften Lauten erkannt wird. In dieser Arbeit wurde ein Algorithmus entwickelt, der Fehler bei der Berechnung der F0 von Sprache detektiert und einen entsprechenden Korrekturvorschlag liefert (erster Durchlauf). Dieser Korrekturvorschlag wird durch eine manuelle Evaluierung bestätigt, ausgebessert oder verworfen. Nach dem manuellen Evaluierungsschritt werden in einem erneuten Durchlauf (zweiter Durchlauf) des Algorithmus die automatisch detektierten sowie die manuell angepassten Korrekturen ausgeführt. Die letztendliche Ausgabe des Algorithmus ist ein Zeit- und Frequenzvektor mit den korrigierten F0-Werten in tabellarischer Form sowie eine Grafik zu jedem korrigierten Berechnungsfehler für die Nachvollziehbarkeit der angewendeten Korrekturen.
Der manuelle Evaluierungsschritt zeigte, dass ein Großteil (90%) der Korrekturvorschläge ohne Änderung übernommen werden konnten. Sofern doch eine Änderung notwendig war, handelte es sich hierbei zumeist nur um kleine Veränderung. Vor allem die Sprunggrößen sowie die Art der detektierten Berechnungsfehler musste angepasst werden. Besonders bei der Detektion von Unstetigkeiten zeichnete sich der Algorithmus aus. Eine stichprobenartige Analyse des F0-Verlaufs in Bereichen, wo keine Berechnungsfehler detektiert wurden, zeigte zumeist einen plausiblen Verlauf der Grundfrequenz. Dies legt den Schluss nahe, dass der Algorithmus eine hohe Präzision aufweist. Es wurden nur wenige F0-Berechnungs-Fehler übersehen und nur wenige Stellen fälschlicherweise als Fehler deklariert.
Herausfordernd war es besonders wenn zwei Fehler von verschiedener Art zeitlich sehr dicht beieinander lagen. Hier mussten, um eine klare Trennung zu erzeugen, die meisten manuellen Korrekturen durchgeführt werden. Wie aus den Ergebnissen der Evaluierung zu erkennen ist, ist dieser Algorithmus für knarrige Stimmen, wie bspw. die von älteren Personen (wie 039F), womöglich nicht geeignet. Ebenso kann eine Fehl-F0, die im typischen Frequenzbereich eines:r Sprechers:in liegt, nicht automatisch erkannt werden. Hierfür ist der manuelle Evaluierungsschritt unerlässlich.
Dies könnte in zukünftigen Arbeiten durch Segmentierung der einzelnen Laute gelöst werden. Prinzipiell könnte durch das Wissen, welche Laute gesprochen werden, sofort eine Aussage darüber getroffen werden, ob es eine F0 geben kann oder nicht. Weiß man z.B., dass gerade der Vokal [a] gesprochen wird, so ist mit hoher Wahrscheinlichkeit eine F0 vorhanden. Wenn hingegen ein stimmloser Frikativ, z.B. [s] gesprochen wird, kann man mit Sicherheit sagen, dass keine F0 vorhanden sein kann. Zwar ließe sich die korrekte Detektion einer Fehl-F0 vermutlich in den meisten Fällen sicherstellen, jedoch erfordert die Segmentierung der einzelnen Laute eine aufwendige Vorverarbeitung.
Die Menge der gefundenen Berechnungsfehler macht die Notwendigkeit dieser Arbeit deutlich, da besonders in der Phonetik ein exakter F0-Verlauf für prosodische Analysen notwendig ist. Aber auch in anderen Bereichen der Sprachsignalverarbeitung wie der automatischen Spracher-

kennung spielen Berechnungsfehler eine wichtige Rolle. Vor allem bei veränderter, nicht-modaler Stimmqualität werden die Laute oftmals falsch erkannt und sogar auch falsch segmentiert. Dies kann wiederum dazu führen, dass ganze Worte anders aufgeteilt oder komplett falsch erkannt werden.
In dieser Arbeit wurde gezeigt, dass der überwiegende Großteil der durch das entwickelte Tool erstellten Korrekturvorschläge bereits ohne Änderungen für die Korrektur des F0-Verlaufs verwendet werden konnten. In einigen Fällen war jedoch eine manuelle Ausbesserung der Korrekturvorschläge notwendig. Durch die Kombination automatischer und manueller Schritte, konnte ein Arbeitsablauf entwickelt werden, der eine schnelle und dennoch präzise Überprüfung und Korrektur an F0-Verläufen ermöglicht. Diese F0-Verläufe können als verlässliche Eingangsdaten einer prosodischen Analyse herangezogen werden.

Literatur

[1] H. G. Tillmann und P. Mansell, *Phonetik: lautsprachl. Zeichen, Sprachsignale u. lautsprachl. Kommunikationsprozess*. Klett-Cotta, 1980.

[2] J. C. Wells, D. Gibbon, R. Moore und R. Winski, „SAMPA computer readable phonetic alphabet," *Handbook of standards and resources for spoken language systems*, Jg. 4, S. 684–732, 1997.

[3] Y. Takefuta, E. G. Jancosek und M. Brunt, „A statistical analysis of melody curves in the intonation of American English," in *Proceedings of the seventh International Congress of Phonetic Sciences/Actes du Septième Congrès international des sciences phonétiques*, De Gruyter Mouton, 2017, S. 1035–1039.

[4] E. Pépiot, „Male and female speech: a study of mean f0, f0 range, phonation type and speech rate in Parisian French and American English speakers," in *Speech Prosody 7*, 2014, S. 305–309.

[5] H. Traunmüller und A. Eriksson, „The frequency range of the voice fundamental in the speech of male and female adults," *Unpublished manuscript*, 1995.

[6] P. A. Keating und C. Esposito, „Linguistic voice quality," *UCLA Working Papers in Phonetics*, Jg. 105, Nr. 105, S. 85–91, 2007.

[7] P. A. Keating, M. Garellek und J. Kreiman, „Acoustic properties of different kinds of creaky voice.," in *ICPhS*, Bd. 2015, 2015, S. 2–7.

[8] K. Kasi und S. A. Zahorian, „Yet another algorithm for pitch tracking," in *2002 ieee international conference on acoustics, speech, and signal processing*, IEEE, Bd. 1, 2002, S. I–361.

[9] L. Ardaillon und A. Roebel, „Fully-convolutional network for pitch estimation of speech signals," in *Insterspeech 2019*, 2019.

[10] D. Talkin und W. B. Kleijn, „A robust algorithm for pitch tracking (RAPT)," *Speech coding and synthesis*, Jg. 495, 1995.

[11] A. M. Noll, „Cepstrum pitch determination," *The journal of the acoustical society of America*, Jg. 41, Nr. 2, S. 293–309, 1967.

[12] Y. Xu und X. Sun, „Maximum speed of pitch change and how it may relate to speech," *The Journal of the Acoustical Society of America*, Jg. 111, S. 1399–1413, Apr. 2002. DOI: 10.1121/1.1445789.

[13] P. Boersma und V. Van Heuven, „Speak and unSpeak with PRAAT," *Glot International*, Jg. 5, Nr. 9/10, S. 341–347, 2001.

[14] B. Schuppler, M. Hagmüller und A. Zahrer, „A corpus of read and conversational Austrian German," *Speech Communication*, Jg. 94, Sep. 2017. DOI: 10.1016/j.specom.2017.09.003.

[15] G. Van Rossum und F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009, ISBN: 1441412697.

[16] Y. Jadoul, B. Thompson und B. de Boer, „Introducing Parselmouth: A Python interface to Praat," *Journal of Phonetics*, Jg. 71, S. 1–15, 2018. DOI: https://doi.org/10.1016/j.wocn.2018.07.001.

[17] H. Buschmeier und M. Włodarczak, „TextGridTools: A TextGrid Processing and Analysis Toolkit for Python," in *Proceedings der 24. Konferenz zur elektronischen Sprachsignalverarbeitung*, Bielefeld, Germany, 2013, S. 152–157.

[18] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke und T. E. Oliphant, „Array programming with NumPy,“ *Nature*, Jg. 585, Nr. 7825, S. 357–362, Sep. 2020. DOI: 10.1038/s41586-020-2649-2. Adresse: https://doi.org/10.1038/s41586-020-2649-2.