



„klarText“ - Barrierefreies Lernen

Ein Projekt im Fachgebiet „Arbeitswelt“ von David Schwendemann (18), Bernau am Chiemsee

Kurzfassung:

Das Ziel des Projekts ist es, das Erlernen und Beherrschen einer Sprache drastisch zu vereinfachen. Durch KI sollen sowohl Texte aus journalistischen Medien als auch Schulbücher und weitere für Deutschlernende und Menschen mit geringen Deutschkenntnissen komplex erscheinende Texte erschließbar werden.

Zudem wurde ergänzend im Zuge des Projektes an einem KI-Modell für die Erkennung und Klassifizierung korrekter Aussprache geforscht.

Hierfür wurde mit Hilfe von Angular und Python eine Web-App programmiert, welche jene Texte in Leichte Sprache übersetzt. Ein KI-Programm überprüft zusätzlich bei lautem Vorlesen das Gesagte und weist den Sprechenden auf Fehler hin.

Es stellt sich folgende Forschungsfrage:

Inwiefern kann der Einsatz von Künstlicher Intelligenz zur Verbesserung der sprachlichen Inklusion von nicht oder unzureichend deutschsprechenden Menschen beitragen?

In dem Projekt werden folgende Aspekte beachtet:

- Entwicklung einer browserbasierten, datenschutzfreundlichen Anwendung
- Umwandlung komplexer Texte in Leichte Sprache mithilfe von KI
- Analyse der Aussprache beim lauten Vorlesen mit automatischem Feedback

Inhaltsverzeichnis

| | |
|--|----|
| Kurzfassung | 1 |
| 1. Einleitung | 3 |
| 1.1 Ideenfindung..... | 3 |
| 1.2 Realisationsansatz der Lösung | 3 |
| 2. Hintergrund und theoretische Grundlagen | 4 |
| 2.1 Leichte Sprache - Regeln und Standards | 4 |
| 2.2 Grundlagen der Spracherkennung und Phonetik | 5 |
| 3. Vorgehensweise, Materialien und Methoden..... | 6 |
| 3.1 Systemarchitektur und technischer Aufbau..... | 6 |
| 3.1.1 Frontend..... | 6 |
| 3.1.2 Backend | 6 |
| 3.2 KI-Modell und Evaluation der Ausgabe..... | 7 |
| 3.2.1 Eingabeverarbeitung | 7 |
| 3.2.2 UI-basierte Verarbeitung | 8 |
| 3.3 Umsetzung der KI-Ausspracheanalyse..... | 8 |
| 3.3.1 Voraussetzung & Eingabeverarbeitung | 9 |
| 3.3.2 Phoneme: Aufspaltung & Erkennung..... | 9 |
| 3.3.3 Bewertung der Aussprache und Fehleranalyse | 10 |
| 3.4 Herausforderungen und Lösungen während der Entwicklung | 12 |
| 4. Ergebnisse | 13 |
| 4.1 Beispiele für Textvereinfachungen..... | 13 |
| 4.2 Ergebnisse der Ausspracheanalyse | 14 |
| 5. Ergebnisdiskussion | 14 |
| 5.1 Grenzen des KI-Ansatzes | 14 |
| 5.2 Ethische und gesellschaftliche Aspekte..... | 15 |
| 6. Fazit und Ausblick..... | 16 |
| 6.1 Mögliche zukünftige Weiterentwicklung | 16 |
| 6.2 Antwort auf die Forschungsfrage und Fazit | 16 |
| 7. Literaturverzeichnis..... | 18 |
| 8. Abbildungsverzeichnis | 20 |

1. Einleitung

1.1 Ideenfindung

Durch eine Vielzahl an Krisen wie Kriege oder Umweltkatastrophen nimmt die Migration stark zu [1]. Dieser Anstieg innerhalb der letzten zehn Jahre kann vor allem für Europa als sehr große Chance für den Arbeitsmarkt gesehen werden. Durch Fachkräftemangel und den demographischen Wandel ist Europa, aber insbesondere Deutschland auch mittel- bis langfristig auf Arbeitende aus dem Ausland angewiesen [2]. Das hierbei auftretende Problem ist allerdings oft der Mangel an guten Sprachkenntnissen, welche das Eintreten in den Arbeitsmarkt sowohl für Arbeitgeber als auch Arbeitnehmer vereinfachen würden [3]. Das Projekt setzt hierbei an zwei zentralen Punkten an. Oft ist das Verstehen komplexer Texte sogar für Muttersprachler anspruchsvoll. Da ist offensichtlich, dass schwierige Zusammenhänge, wie sie unter anderem auf behördlichen Webseiten, in juristischen Zusammenhängen oder in ganz normalen Zeitungsartikeln auffindbar sind, für Nichtmuttersprachler besondere Hürden darstellen.

Als zweiten Ansatzpunkt wurde die Aussprache aufgegriffen. Das Erlernen korrekter Aussprache ist aufgrund der hohen Nachfrage an Kursen und dem damit verbundenen Aufwand oft nicht möglich und gerät eher in den Hintergrund [3]. Da eine gute Aussprache allerdings sehr oft große Vorteile auf dem Arbeitsmarkt mitbringt, war hier die Überlegung, eine einfachere und skalierbarere Lösung zu finden.

1.2 Realisationsansatz der Lösung

Von Anfang an war klar, man braucht eine Softwarelösung, wenn man das Problem kurzfristig angehen möchte. Obwohl das Erlernen von Sprache durch direkte menschliche Interaktion aus sozialer Sicht nachhaltiger wäre, [4], können wir an der derzeitigen Situation der hohen Nachfrage nichts ändern, weshalb eine Softwarelösung aufgrund ihrer guten Skalierbarkeit die bessere Wahl ist. Zudem ist die derzeit starke Entwicklung von zahlreichen KI-Modellen ein gutes Potenzial für diese Anwendung. Aus diesen Gründen habe ich mich für die Entwicklung einer Web-App mit Python als Backend entschieden. Durch die Web-App ist ein theoretischer Zugriff auf allen browserfähigen Geräten gewährt. Ein weiterer wichtiger Aspekt für diesen Tech-Stack waren die bereits im Vorhinein vorhandenen Erfahrungen in den genannten Bereichen.

Um Entwicklungskosten zu reduzieren, wurden alle KI-Modelle lokal auf „normaler“ Hardware installiert. Dadurch musste nebenbei auch indirekt darauf geachtet werden, Anfragen und Berechnungen aufgrund mangelnder professioneller Hardware möglichst effektiv zu gestalten, was präventiv auch positive Folgen hat, da dann der Code auf KI spezialisierter Hardware noch effizienter laufen kann.

2. Hintergrund und theoretische Grundlagen

Um die entwickelte Anwendung und deren Funktionsweise besser einordnen zu können, werden in diesem Kapitel die notwendigen theoretischen Grundlagen erläutert. Dabei wird zunächst auf das Konzept der Leichten Sprache eingegangen, da diese einen zentralen Bestandteil der Textvereinfachung darstellt. Anschließend werden grundlegende Aspekte der Spracherkennung sowie der Phonetik beschrieben, welche für die Umsetzung der automatischen Ausspracheanalyse relevant sind.

2.1 Leichte Sprache – Regeln und Standards

Leichte Sprache ist eine stark vereinfachte Form der deutschen Sprache, deren Ziel es ist, Texte möglichst verständlich und barrierearm zu gestalten. Sie richtet sich insbesondere an Menschen mit Lernschwierigkeiten, geringen Sprachkenntnissen oder nicht-muttersprachlichem Hintergrund. Deshalb findet sie vermehrt auch Anwendung im öffentlichen und digitalen Bereich. Gerade bei behördlichen oder informativen Texten kann Leichte Sprache dazu beitragen, komplexe Inhalte verständlicher aufzubereiten und Zugangshürden zu reduzieren [5].

Die Leichte Sprache folgt klar definierten Regeln, die sich sowohl auf die Wortwahl und den Satzbau als auch auf die Struktur eines Textes beziehen. Maßgeblich sind hierbei unter anderem die Empfehlungen des Netzwerks Leichte Sprache [6] sowie die Norm DIN SPEC 33429 [7] [8], welche Kriterien für verständliche Texte festlegt. Diese Regeln dienen als normative Orientierung für die Erstellung und Bewertung von Texten in Leichter Sprache.

Zentrale sprachliche Regeln der Leichten Sprache sind unter anderem:

- Verwendung kurzer und einfacher Sätze, die in der Regel nur eine Aussage enthalten
- Vermeidung von Nebensätzen, Passivkonstruktionen sowie komplexen Zeitformen
- Nutzung bekannter und alltäglicher Wörter anstelle von Fachbegriffen
- Erklärung unvermeidbarer Fremd- oder Fachwörter in einfacher Form
- Verzicht auf Abkürzungen, Metaphern und Ironie, da diese missverständlich sein können

Neben diesen sprachlichen Aspekten spielt auch die Struktur eines Textes eine wichtige Rolle. Inhalte sollen logisch aufgebaut und klar gegliedert sein. Überschriften müssen eindeutig formuliert werden, und Aufzählungen werden bevorzugt eingesetzt, um Informationen übersichtlich darzustellen. Dadurch wird es Leserinnen und Lesern erleichtert, den Inhalt schrittweise zu erfassen [6].

Für die automatische Umsetzung von Leichter Sprache mithilfe von KI stellen diese Regeln eine besondere Herausforderung dar. Natürliche Sprache ist häufig mehrdeutig und kontextabhängig, wodurch eine vollständige regelkonforme Vereinfachung nicht immer eindeutig möglich ist [6]. Ziel ist es daher, Texte so zu vereinfachen, dass sie besser verständlich sind, ohne dabei wesentliche Inhalte zu verlieren. Die in dieser Arbeit entwickelte Lösung orientiert sich an den genannten Regeln und nutzt diese als Grundlage für die Bewertung der erzeugten Textausgaben.

2.2 Grundlagen der Spracherkennung und Phonetik

Die automatische Spracherkennung beschäftigt sich mit der Verarbeitung und Analyse gesprochener Sprache durch Computersysteme. Ziel ist es, gesprochene Wörter in Text umzuwandeln oder deren Aussprache zu bewerten. Solche Technologien werden heute bereits in vielen Bereichen eingesetzt, beispielsweise in Sprachassistenten, Übersetzungsprogrammen oder Lernanwendungen [9]. Der Prozess der Spracherkennung beginnt mit der Aufnahme eines Audiosignals. Dieses Signal wird anschließend vorverarbeitet, um Störgeräusche zu reduzieren und relevante Merkmale der Sprache herauszufiltern. Moderne Systeme wie jenes, welches im Rahmen dieses Praxisprojekts verwendet wurde, nutzen hierfür meist KI-Modelle, die auf große Mengen an Sprachdaten trainiert wurden [9]. Diese Modelle sind in der Lage, gesprochene Sprache auch bei unterschiedlichen Stimmen, Akzenten oder Sprechgeschwindigkeiten zu erkennen. Ein wichtiger Bestandteil der Spracherkennung ist die Phonetik. Die Phonetik beschäftigt sich mit den Lauten der Sprache und deren Bildung [9]. Die kleinsten bedeutungsunterscheidenden Laute werden als Phoneme bezeichnet und im Internationalen Phonetischen Alphabet (kurz IPA) dokumentiert [10]. Für die deutsche Sprache sind dies beispielsweise Laute wie /b/, /p/, /ø/ oder /ʃ/. Eine korrekte Aussprache eines Wortes setzt sich aus der richtigen Abfolge und Artikulation dieser Phoneme zusammen.

Für eine automatische Ausspracheanalyse wird gesprochene Sprache in ihre einzelnen Phoneme zerlegt und mit einer Referenzaussprache verglichen. Dabei kann festgestellt werden, ob bestimmte Laute falsch ausgesprochen, ausgelassen oder durch andere ersetzt wurden. Auf diese Weise ist eine deutlich genauere Analyse möglich, als wenn lediglich das gesamte Wort betrachtet wird [9]. Im Rahmen dieser Arbeit bildet die phonemische Analyse die Grundlage für die Bewertung der Aussprache. Ziel ist es, nicht nur zwischen richtig und falsch zu unterscheiden, sondern konkrete Hinweise auf Aussprachefehler geben zu können. Dies ist insbesondere für Lernende von Vorteil, da gezielt an einzelnen Lauten gearbeitet werden kann. Die beschriebenen Grundlagen der Spracherkennung und Phonetik sind somit essenziell für das Verständnis der technischen Umsetzung der entwickelten Anwendung.

3. Vorgehensweise, Materialien und Methoden

3.1 Systemarchitektur und technischer Aufbau

Da das Projekt möglichst viele Menschen erreichen sollte, wurde sich für eine Web-App entschieden. Diese ist auf nahezu allen modernen internetfähigen Geräten plattformübergreifend aufrufbar. Zusätzlich ergibt sich der Vorteil, das Projekt nur einmal programmieren zu müssen, ohne es mühevoll an die Programmiersprachen der jeweiligen Betriebssysteme anzupassen. Das Frontend der Web-App kann auf einem Server gehostet werden und anschließend per Domain aufgerufen werden.

Um die Zugänglichkeit noch weiter zu erhöhen, wurde zudem auf ein responsives Design geachtet. In der Webentwicklung spricht man davon, wenn sich die Webseite dynamisch an die Displaygröße des Endgerätes anpasst.

Die erstellte Web-App muss im Hintergrund bei der Text- und Sprachverarbeitung eine große Anzahl an Daten speichern und verarbeiten. Dies geschieht nicht im Frontend, auf dem Endnutzergerät, sondern wird vom Backend übernommen. Mittels APIs und HTTP(S)-Requests ist das Frontend dann, bei Aufruf einer Anfrage im Austausch mit dem Backend.

3.1.1 Frontend

Heutzutage werden alle Webseiten mit HTML und CSS erstellt [11]. Diese sind technisch gesehen keine Programmiersprachen, sondern Stylesheet-Sprachen. Sie sorgen für eine angenehme User-Experience.

Für die Logik im Frontend wurde in der Entwicklung die Sprache TypeScript (TS) verwendet. Im Vergleich zum Vorgänger JavaScript, bietet TypeScript eine statische Typisierung, um Fehler frühzeitig zu erkennen und die Code-Qualität zu verbessern [12].

Für eine angenehmere Entwicklungserfahrung wurde das von Google entwickelte Web-Framework Angular verwendet. Frameworks dienen als Software-Grundgerüst mit vorgefertigten Werkzeugen, Bibliotheken und Strukturen, welche die Entwicklung von Websites, Web-APIs und Webanwendungen vereinfacht und beschleunigt [13].

3.1.2 Backend

Das Backend sorgt serverseitig für eine korrekte Verarbeitung der Daten und eingehenden Abfragen. In unserem Fall wurde das Backend in Python programmiert. Python ist eine vielseitige Programmiersprache, die aufgrund ihrer klaren Syntax und umfangreichen Dokumentation häufig in der Forschung für Datenanalyse und im Bereich KI eingesetzt wird [14]. Gerade letzteres war für dieses

Projekt von hoher Relevanz. Eingehende Anfragen werden an die jeweiligen Funktionen weitergeleitet und dort (meist asynchron) verarbeitet.

In Python wird zudem noch die Paketverwaltung „conda“ genutzt. Sie ist im Gegensatz zu der gängigen Standardpaketverwaltung „pip“ eine gute Alternative aufgrund ihrer genaueren Prüfung der Paketabhängigkeiten [15].

Die verwendeten KI-Modelle und deren zugehörigen Python-Bibliotheken laufen nur auf Linux-basierten Betriebssystemen. Aus diesem Grund wurde das Backend auf einem Ubuntu Server gestartet.

3.2 Umsetzung & Evaluation des LLM und der KI-Textanalyse

Um die Textanalyse konkret realisieren zu können, wurde mithilfe von Ollama ein KI-Modell lokal installiert. Dadurch wird zwar eine lokale Nutzung, allerdings auf eigene Ressourcen ermöglicht. Die lokale Nutzung ermöglicht eine vollständige Kontrolle über die Anpassung und Feinabstimmung des Modells [16].

3.2.1 Eingabeverarbeitung

Vor der Verarbeitung wird die Eingabe des Nutzers im Backend validiert und angepasst. Um die Eingaben auch auf leistungsschwacher Hardware bewältigen zu können, wird die Anfrage nicht als Ganzes gestellt, sondern in 3 Phasen aufgeteilt. Für große LLMs mit >100B Parametern, die auf JSON ausgaben spezialisiert sind und allgemein ein gutes Sprachverständnis haben, stellt das Senden einer großen Anfrage keinerlei Problem dar. Da unser Modell allerdings lokal laufen muss, gilt es eine effiziente Verteilung der Anfragen zu priorisieren.

In der ersten Phase wird versucht, eine möglichst kontextnahe Vereinfachung des Originaltextes zu erstellen. Die hierfür gestellte Anfrage beinhaltet einen aussagekräftigen Prompt und den Inhalt des Textes. Diese Vereinfachung stellt das Kernstück der Leichten Sprache dar. Der Prompt ist mit den Regeln von Leichter Sprache versehen, um die Ausgabe verständlich, klar und barrierearm zu gestalten.

In Phase zwei wird dann ergänzend versucht schwierige Wörter aus dem Originaltext zu filtern. Die hierfür gestellte Anfrage sucht im Originaltext komplexe (Fach-) Begriffe und versucht eigenständig dafür geeignete Begriffserklärungen zu finden. Diese werden dann später für den Nutzer angezeigt, wenn dieser den Mauszeiger über den markierten Fachbegriff bewegt.

Durch diesen Schritt soll ein Lerneffekt erzielt werden. Der Nutzer hat dann nicht nur den Text als vereinfachten Fließtext, sondern ebenfalls einfache Erklärungen bestimmter Begriffe, um sein Verständnis über den Text zu vertiefen.

3.2.2 UI-basierte Verarbeitung

Neben den zahlreichen Regeln, die sich auf den Inhalt des Textes beziehen, gibt es auch einige Vorgaben, welche die Darstellung, die Typography und das visuelle Design des Textes regulieren [6]. Hierzu zählen unter anderem:

- Verneinungen immer **fett**
- Schriftart: gerade Schrift oder einfache Handwritten-Sans
- angemessene Schriftgröße & Zeilenabstand $\geq 1,5$
- Text immer linksbündig
- Sätze immer in eine neue Zeile & keine Worttrennungen zwischen Zeilen
- Kontrast zwischen Hintergrund und Text

Entscheidend für die richtige Anzeige ist bereits die korrekte Formatierung in ein JSON-Format. Dieses trennt Sätze, um die zeilenspezifisch anzuzeigen immer nach einem Satzende. Ebenso werden Verneinungen schon im Voraus erkannt und speziell markiert durch voranstehende „***“ Zeichen. Auf Grundlage dieser strukturierten Informationen ist das Frontend in der Lage, die definierten Darstellungsregeln zuverlässig umzusetzen. Ergänzende Gestaltungsvorgaben, wie Schriftart, Ausrichtung oder Kontrast, werden durch einfache CSS-Formatierungen umgesetzt.

Auch die Begriffserklärungen auf der Originaltext-Seite benötigen eine spezielle Formatierung. Hier würde sich für den Einsatz von Tooltip-Elementen entschieden. Dabei werden einzelne Wörter im Text hervorgehoben und können beim Darüberfahren mit der Maus zusätzliche Erklärungen anzeigen. Diese Methode ermöglicht es, wichtige Informationen kontextsensitiv zu vermitteln, ohne den Lesefluss zu unterbrechen, und unterstützt gleichzeitig die visuelle Klarheit des Textes.

Technisch gesehen wird dies über eine TypeScript-Funktion umgesetzt, die den Text auf von der KI gegebene Ausdrücke analysiert, diese findet und sie dynamisch in HTML-Elemente einbettet. Diese Elemente kombinieren die visuelle Hervorhebung der Wörter mit einem Tooltip, der die Zusatzinformationen enthält. Durch diese Vorgehensweise lassen sich die Gestaltungsvorgaben konsistent umsetzen und die Interaktivität der Benutzeroberfläche sicherstellen.

3.3 Umsetzung der KI-Ausspracheanalyse

Um die gesprochene Eingabe des Nutzers korrekt in Phoneme verarbeiten zu können und im Anschluss die dadurch gesammelten Daten richtig interpretieren zu können, ist ein komplexer Prozess nötig, welcher in mehrere Module unterteilt werden kann.

3.3.1 Voraussetzung und Eingabeverarbeitung

Die Eingabe erfolgt über das Frontend. Nach erfolgreicher Aufnahme kann die Audiodatei an das Backend gesendet werden. Je nach Audio-Library oder Browser wird die Datei in unterschiedlichen Formaten an das Backend gesendet. Da dort zwingend eine Datei im Waveform Audio File Format (WAVE) benötigt wird, muss diese gegebenenfalls noch konvertiert werden [17]. Im Fall des Projektes wird eine MP3 Datei an das Backend gesendet und dort in WAVE umgewandelt.

Zusätzlich muss die Abtastrate auf 16.000 Hz gesetzt werden. Dies ist der Standard für die meisten KI-Modelle [18]. Zugleich wird das Audio, falls es Stereo ist, zu einem Kanal zusammengefasst. Mithilfe der Python Bibliothek „librosa“ wird das Audio noch zusätzlich modifiziert, indem die Lautstärke der Eingabe normalisiert wird, um große Tondifferenzen zu vermeiden. Zusätzlich wird die Datei noch mit einem Universally Unique Identifier (UUID) gespeichert [19].

3.3.2 Phoneme: Aufspaltung und Erkennung

Zur erfolgreichen Klassifizierung der Aussprache muss das Gesprochene vorerst in seine einzelnen Phoneme gespalten werden. Der hierfür nötige Prozess kann von einem KI-Modell übernommen werden. In diesem Fall wurde der „Montreal Forced Aligner (MFA)“ verwendet, da dieser dank seiner einfachen Installation über conda und seinem universellen Zugriff über die Befehlsleiste die Einrichtung sehr erleichtert hat [20].

MFA braucht, um Phoneme in einem Audio zu erkennen zwingend eine Transkription des Gesprochenen. Diese wird noch vor der eigentlichen Extraktion von dem ebenfalls lokal installierten general-purpose speech recognition Modell Whisper von OpenAI erstellt. Die hierbei entstandene Datei muss namensgleich wie das Audio am selben Ort gespeichert werden, um im Anschluss von MFA verarbeitet werden zu können [20].

```
32      # Alignment ausführen
33      run_align_corpus(
34          corpus_directory=temp_dir,           # enthält Audio + Transcript
35          dictionary_path="german.dict",     # MFA German dictionary
36          acoustic_model_path="german.zip",  # Pfad zum deutschen Akustikmodell
37          output_directory=temp_dir,
38          config_path=None
39      )
40
```

Abbildung 1: Beispielhafter Aufruf des Montreal Forced Aligners

In Abbildung 1 ist der nötige Befehl zum Aufruf des MFA KI-Modells zu sehen.

Für die Verarbeitung nötig sind [21]:

- der Input-Dateipfad zur WAVE- und Textdatei
- ein für die jeweilige Sprache verwendete Dictionary, welches sprachzugehörige Phoneme beinhaltet
- das Akustikmodell der jeweiligen Sprache
- optional eine Konfigurationsdatei (hier nicht verwendet)

Der Aufruf gibt eine TextGrid-Datei wieder, welche die Grundlage für die Phonemanalyse bildet. Aus ihr werden die relevanten Phonemintervalle extrahiert. Jedes Phonem wird zusammen mit seinen Start- und Endzeiten in ein maschinenlesbares JSON-Format überführt. Diese Struktur ermöglicht die präzise Zuordnung der Audiosegmente zu den einzelnen Phonemen für die GOP-Berechnung.

Eine mögliche Ausgabe im JSON-Format für ein Phonem ist beispielsweise folgende:

```
{  
    "phoneme": "\u0259",  
    "start": 0.04,  
    "end": 0.15,  
    "gop_score": -26.8306  
},
```

Abbildung 2: JSON-Ergebnis, Phonem & GOP Score

3.3.3 Bewertung der Aussprache und Fehleranalyse

Nach der erfolgreichen Phonemextraktion erfolgt die Bewertung der Aussprache mittels des **Goodness of Pronunciation (GOP)**-Verfahrens [22]. GOP quantifiziert, wie gut ein Sprecher ein Phonem im Vergleich zu einem Referenzmodell artikuliert. Höhere Werte deuten auf eine korrekte Aussprache hin, während niedrigere Werte auf Abweichungen oder Unsicherheiten schließen lassen. Hierfür wird ein Wav2Vec2 Modell verwendet.

Der gesamte Analyseprozess lässt sich wiederum in einzelne Schritte unterteilen: GOP-Berechnung, Speicherung der Ergebnisse und automatische Fehleranalyse mit Feedbackgenerierung.

Das vorverarbeitete Audio wird durch das **Wav2Vec2-Modell** geleitet, das für jedes Zeitintervall Wahrscheinlichkeiten für alle möglichen Phonem-Labels (Logits) erzeugt. Für jedes Phonem aus der MFA-Extraktion werden die entsprechenden Logits über die Dauer des Phonems extrahiert. Dabei wird die Zeit in Frame-Indizes umgerechnet, um die Synchronität zwischen Audio und Phonemintervallen

sicherzustellen. Informationen hierfür hat das Modell über die vorangegangene Phonemextraktion, bei der jeweils Start- und Endzeitpunkt, wie in **3.3.2** beschrieben, ebenfalls gespeichert werden.

Der GOP-Score wird mittels folgender (vereinfachter) Formel berechnet [23]:

$$GOP = \log P(\text{target}) - \max \log P(\text{andere})$$

Hierbei entspricht $P(\text{target})$ der mittleren Log-Wahrscheinlichkeit des korrekten Phonems, und $\max P(\text{andere})$ der maximalen Log-Wahrscheinlichkeit konkurrierender Phoneme. Dieser Wert erlaubt die quantitative Bewertung der Aussprachequalität jedes Phonems.

Phoneme, die nicht im Vokabular des Modells enthalten sind, werden mit None markiert, um fehlerhafte oder unbekannte Einträge zu kennzeichnen.

Auf Grundlage der berechneten GOP-Scores wird eine differenzierte Analyse der Aussprache durchgeführt [22]. Zunächst erfolgt eine Klassifizierung der einzelnen Phoneme anhand definierter Schwellenwerte, wodurch jedes Phonem in die Kategorien „gut“, „mittel“ oder „schwach“ eingeordnet wird. Anschließend werden statistische Kennwerte für jedes Phonem ermittelt; insbesondere wird der Mittelwert der GOP-Scores berechnet und zusätzlich auf eine 0–100-Skala transformiert, um eine leicht interpretierbare Darstellung der Aussprachequalität zu ermöglichen.

Auf Basis dieser Bewertungen werden phonematische Stärken und Schwächen identifiziert. Phoneme mit hohen Übereinstimmungswerten gegenüber dem Referenzmodell werden als Stärken ausgewiesen, während Phoneme mit niedrigen GOP-Wahrscheinlichkeiten als Schwächen klassifiziert werden [23]. Für die als schwach bewerteten Phoneme werden gezielte Trainingsmaßnahmen vorgeschlagen, darunter spezifische Übungen wie Minimalpaare sowie artikulatorische Hinweise. Diese Empfehlungen umfassen Aspekte der Lautbildung, Zungen- und Lippenposition sowie Atemführung, jeweils angepasst an die charakteristischen Eigenschaften des betreffenden Phonems.

Die Ergebnisse der Analyse werden in mehreren Formaten aufbereitet, um unterschiedliche Nutzungsszenarien abzudecken. Eine strukturierte JSON-Datei enthält detaillierte Informationen zu Stärken, Schwächen, Übungsanleitungen und einer Heatmap, die die zeitliche Verteilung der GOP-Scores visualisiert. Zusätzlich wird eine Textdatei erzeugt, die eine kompakte Übersicht der Ergebnisse für Debugging oder Export bereitstellt. Schließlich wird ein Frontend-kompatibles JSON erstellt, das

die Gesamtbewertung, phonemspezifische Scores und Übungsanweisungen integriert und somit die unmittelbare Verwendung in Benutzeroberflächen ermöglicht.

Die Heatmap visualisiert die Aussprachequalität jedes Phonems auf einer Skala von 0 bis 100, wodurch problematische Abschnitte der Audioaufnahme identifiziert und gezielt trainiert werden können. Auf diese Weise erlaubt die Methodik eine präzise, reproduzierbare und anwendungsorientierte Analyse der Aussprache.

3.4 Herausforderungen und Lösungen während der Entwicklung

Während der Entwicklung des gesamten Minimum-Viable-Product (MVP) gab es einige Komplikationen. Insbesondere die Arbeit mit KI hat einige Schwierigkeiten mit sich gebracht.

Beim Modell „Sprache vereinfachen“ war eines der größten Probleme, lange Eingaben samt ausführlichem Prompt bei angemessener Laufzeit mit möglichst gutem KI-Modell zu ermöglichen. Bei großen Modellen, also z.B. KI-Modelle mit >30B (Billion, dt: Milliarde) Parametern waren die Ergebnisse deutlich zuverlässiger. Vor allem das Experimentieren mit großen, cloudbasierten LLMs wie Googles Gemini hat sehr gute Ergebnisse geliefert, was unter anderem darauf zurückzuführen sein könnte, dass diese laut Schätzungen ~200B bis 1T Parameter haben [24].

Beim selbständigen Testen mit lokal installierten Modellen, wie LLama (Meta) oder das endgültig verwendete Qwen2.5 (7B) musste dies einerseits durch schlaueres prompten und andererseits durch eine Aufteilung des Prozesses in kleinere Unterprozesse gelöst werden. Vor allem letzteres hatte allerdings zur Folge, dass das Sprachmodell den Kontext der Eingabe nicht mehr nachvollziehen konnte. Hier musste ein passender Kompromiss gefunden werden, welcher den Kontext und die Dauer der Anfrage beachtet.

Leichte Sprache hat zudem ein sehr vielschichtiges Regelwerk. Laut dem Netzwerk Leichte Sprache muss diese immer von einem Prüfer oder einer Prüferin gegengelesen werden [6]. Diesen Punkt unterstützt die automatisierte Umsetzung als KI-Modell natürlich nicht.

Auch bei der Analyse der Aussprache traten während der Entwicklungszeit zahlreiche Probleme auf. Ein hier allgemein aufgetretener Konflikt ist die Bewertung von Aussprache allgemein. Aussprache ist nicht immer richtig oder falsch. Es gibt nicht die eine richtige perfekte Aussprache eines Lautes. Manche Wörter können anders ausgesprochen auch richtig sein. Zudem kommt es, dass die Anwendung keinerlei Dialekt beachtet und teilweise auch Unterschiede je nach Sprecher und Artikulation auftreten. Faire Scores zu erzeugen, die die tatsächliche Qualität der Aussprache bewerten, hat sich als schwieriger herausgestellt, als initial vermutet.

Auch diese Scores in ausdrucksstarke Bewertungen umzuwandeln, war anspruchsvoll. Ein technisch

korrektes Feedback muss nicht immer didaktisch hilfreich sein. Selbst wenn in der Theorie eine vollständig richtige Erkennung und Bewertung der Aussprache vorliegen würde, muss das nicht heißen, dass das definierte Feedback dem Lernenden auch wirklich hilft. Die Frage wie das Falsch gesagte verbessert werden kann, wird bisher nur vergleichsweise oberflächlich behandelt.

4. Ergebnisse

Dieses Kapitel zeigt die praktischen Ergebnisse des Projekts. Dabei gehen die Ergebnisse zur automatischen Textvereinfachung in Leichte Sprache genauso mit ein wie die Resultate der KI-gestützten Ausspracheanalyse. Ziel ist, anhand konkreter Beispiele zu zeigen, wie gut die entwickelte Anwendung funktioniert und welchen Nutzen Lernende damit tatsächlich haben.

4.1 Beispiele für Textvereinfachungen

| Eingabe | Ausgabe |
|---|--|
| <p>Düsseldorf. Der US-Elektroautobauer Tesla hat die Batterieproduktion in seiner Gigafabrik in Grünheide wieder aufgenommen. Tesla hatte in Grünheide bereits zuvor Batterien gefertigt. Nach Angaben von Werksleiter Thierig produzierten die Mitarbeiter rund 100.000 Einheiten eines älteren Batterietyps. Anschließend baute der Konzern die Fertigung um und schuf so die Voraussetzungen für einen neuen Batterietyp. Intern firmiert das Vorhaben unter dem Codename „Projekt Coyote“.</p> | <p>Große Produktionsanlage</p> <p>Neue Batterien bei Tesla in Grünheide</p> <p>In Grünheide gibt es eine große Fabrik für Elektro-Autos. Die Fabrik gehört der Firma Tesla aus den USA. In dieser Fabrik baut Tesla jetzt wieder Batterien.</p> <p>Was vorher passiert ist</p> <p>Tesla hat in Grünheide schon früher Batterien gebaut. Die Mitarbeiter haben etwa 100.000 Stück von einer alten Batterie gemacht. Danach hat Tesla die Fabrik umgebaut. Das Ziel war: Die Fabrik soll einen neuen Typ von Batterien bauen können.</p> <p>Das Projekt Coyote</p> <p>Der Chef von der Fabrik heißt Herr Thierig. Er sagt: Der Umbau ist fertig. Tesla hat einen geheimen Namen für diesen Plan. Der Name ist: Projekt Coyote.</p> |

Abbildung 3: Leichte Sprache UI-Ausgabe

In Abbildung 2 ist die finale Nutzerausgabe der Web-App zur Textvereinfachung dargestellt. Auf der linken Seite wird der Originaltext samt Begriffserklärungen angezeigt, während auf der rechten Seite die vereinfachte Version in Leichter Sprache ausgegeben wird. Die Gegenüberstellung ermöglicht es den Nutzenden, den ursprünglichen Inhalt direkt mit der vereinfachten Fassung zu vergleichen und Zusammenhänge besser nachzuvollziehen.

Die vereinfachte Ausgabe folgt konsequent den Regeln der Leichten Sprache. Lange und komplexe Sätze wurden in kurze, klar strukturierte Aussagen zerlegt. Fachbegriffe und abstrakte Formulierungen wurden entweder durch einfachere Wörter ersetzt oder vollständig vermieden. Zusätzlich wurde auf eine klare zeitliche und inhaltliche Abfolge geachtet, sodass der Text auch ohne Vorwissen verständlich bleibt.

Die Ausgabe war bei den verschiedensten Texten konsistent. Es wurden alle Regeln in 18 von 20 getesteten Fällen eingehalten. Ergänzend zeigte sich, dass die verwendete Formatierung stets vorteilhaft war und den Lesefluss verbessert hat.

4.2 Ergebnisse der Ausspracheanalyse

Der folgende JSON-Schnipsel zeigt ein mögliches Feedback:

```
{  
    "headline": "🗣️ Aussprache-Feedback",  
    "overall": {  
        "score": 45.1,  
        "message": "Gute Basis: gezieltes Training bringt Fortschritte."  
    },  
    "strengths": [],  
    "weaknesses": [  
        {  
            "phoneme": "d",  
            "score": 43.5,  
            "tip": "Stimmhaft, Zungenspitze locker an den Zähnen."  
        }  
    ]  
}
```

Abbildung 4: Feedback Ausgabe

Insgesamt wird ein Score von 45,1 erzielt. Dieser kann durch Feinabstimmung von Parametern und Schwellenwerten im Vorhinein beeinflusst werden, ist allerdings bei gleichbleibender Konfiguration immer ähnlich. Bei mehreren Versuchen sind klare Unterschiede erkennbar geworden. Abweichungen des Scores bei absichtlich schlechter Aussprache sind nach eigenen Tests bis +30 Scorerpunkte aufgetreten. Auch die KI-Ausspracheanalyse ist dementsprechend funktionstüchtig.

5. Ergebnisdiskussion

Die Ergebnisse deuten darauf hin, dass der KI-basierte Ansatz grundsätzlich taugt, um Sprachbarrieren zu senken und Lernprozesse zu unterstützen. Trotzdem sollten die Resultate mit Vorsicht betrachtet werden - sowohl technisch als auch gesellschaftlich gilt es, genauer hinzusehen.

5.1 Grenzen des KI-Ansatzes

Trotz der allgemein sichtbaren positiven Ergebnisse sind bei der entwickelten Lösung klare Grenzen sichtbar. Die automatische Umsetzung von Leichter Sprache kann menschliche Prüferinnen und Prüfer

nicht vollständig ersetzen. Dies wird allerdings laut dem Netzwerk für Leichte Sprache dringend empfohlen [6]. Insbesondere bei komplexen oder mehrdeutigen Inhalten besteht die Gefahr, dass wichtige Nuancen verloren gehen oder Inhalte vereinfacht, aber nicht korrekt wiedergegeben werden.

Für den normalen Nutzer sind diese Nuancen wahrscheinlich größtenteils irrelevant. Es wird nur deutlich, dass das Modell nicht für öffentlich einsehbare Texte, wie sie beispielsweise in Museen vorkommen, verwendet werden soll. Der private Nutzer muss sich nur dessen bewusst sein, dass KI immer Fehler machen kann.

Auch die Ausspracheanalyse weist Einschränkungen auf. Die Bewertung basiert auf im Voraus trainierten Modellen, welche nicht alle Varianten korrekter Aussprache abbilden können. Dialekte, individuelle Sprechweisen oder alternative Artikulationen werden nur unzureichend berücksichtigt. Zudem ist die didaktische Qualität des Feedbacks noch ausbaufähig, da technische Genauigkeit nicht automatisch zu optimalem Lernerfolg führt.

Darüber hinaus ist die Leistungsfähigkeit der Anwendung stark von der verfügbaren Hardware abhängig. Es macht sich die lange Berechnungsgeschwindigkeit von über einer Minute bemerkbar. Der lokale Betrieb der KI-Modelle stellt zwar einen Vorteil hinsichtlich des Datenschutzes, begrenzt jedoch die Modellgröße und damit auch die Qualität der Ergebnisse.

5.2 Ethische und gesellschaftliche Aspekte

Der Einsatz von KI im Bildungs- und Sprachlernkontext wirft auch ethische Fragestellungen auf. Positiv ist hervorzuheben, dass die entwickelte Anwendung den Zugang zu Informationen erleichtert und somit einen Beitrag zur Chancengleichheit leisten kann. Insbesondere für Menschen mit Migrationshintergrund oder Lernschwierigkeiten kann dies eine wertvolle Unterstützung darstellen.

Gleichzeitig besteht die Gefahr, dass automatisierte Systeme als Ersatz für menschliche Unterstützung missverstanden werden. Sprachlernen ist ein sozialer Prozess, der durch zwischenmenschliche Interaktion geprägt ist [25]. KI-basierte Anwendungen sollten daher nicht als vollständiger Ersatz, sondern als ergänzendes Hilfsmittel betrachtet werden.

Ein weiterer zentraler Aspekt ist der Datenschutz. Durch den lokalen Betrieb der Modelle werden sensible Text- und Sprachdaten nicht an externe Server übertragen, was einen verantwortungsvollen Umgang mit personenbezogenen Daten ermöglicht. Dennoch ist Transparenz gegenüber den Nutzerinnen und Nutzern notwendig, insbesondere hinsichtlich der Funktionsweise der KI und der Grenzen der automatisierten Bewertungen.

6. Fazit und Ausblick

6.1 Mögliche zukünftige Weiterentwicklung

Die entwickelte Web-App stellt einen funktionsfähigen Prototyp dar, der jedoch in mehreren Bereichen weiter ausgebaut werden kann. Ein zentraler Ansatzpunkt ist die Verbesserung der Textvereinfachung durch feinere Regelmodelle und adaptive Prompt-Strukturen. Insbesondere eine stärkere Berücksichtigung des ursprünglichen Kontexts könnte dazu beitragen, inhaltliche Verluste bei der Vereinfachung weiter zu reduzieren.

Darüber hinaus wäre eine Kombination aus automatischer Vereinfachung und menschlicher Nachprüfung denkbar. Eine solche hybride Lösung würde den Anforderungen des Netzwerks Leichte Sprache besser entsprechen und gleichzeitig den Aufwand für Prüferinnen und Prüfer deutlich verringern [6]. Auch eine Erweiterung auf weitere Sprachen oder Sprachstufen, etwa Einfache Sprache, stellt eine mögliche Weiterentwicklung dar.

Im Bereich der Ausspracheanalyse besteht weiteres Potenzial in der Verbesserung der didaktischen Rückmeldungen. Zukünftig könnten personalisierte Trainingspläne erstellt werden, die sich dynamisch an die individuellen Schwächen der Lernenden anpassen. Ebenso wäre eine stärkere Berücksichtigung von Dialekten und akzeptierten Aussprachevarianten sinnvoll, um die Bewertung fairer und robuster zu gestalten. Genauso wie das Vereinfachungs-Modul hat auch die Ausspracheanalyse das Potenzial auf mehrere Sprachen ausgeweitet zu werden. Dazu kann man die Ausspracheanalyse noch vielseitig gamifizieren. In der realen Welt kann das eine höhere Nutzerakzeptanz zur Folge haben.

Technisch könnte die Anwendung durch den Einsatz leistungsfähigerer Hardware oder spezialisierter KI-Beschleuniger effizienter betrieben werden. Alternativ wäre auch ein optionaler Cloud-Betrieb denkbar, um größere Modelle mit höherer Genauigkeit einsetzen zu können, sofern datenschutzrechtliche Aspekte berücksichtigt werden.

6.2 Antwort auf die Forschungsfrage und Fazit

Die zentrale Fragestellung des Projekts lautete, ob und wie der Einsatz von KI für eine bessere Inklusion für nicht- oder schlecht Deutsch-Sprechende genutzt werden kann. Auf Grundlage der durchgeföhrten Implementierung und Evaluation lässt sich diese Frage grundsätzlich mit „ja“ beantworten.

Die Ergebnisse zeigen, dass KI-Modelle in der Lage sind, Texte korrekt und nach vorgegebenen Regeln zu vereinfachen, um dadurch sprachliche Barrieren zu reduzieren. Ergänzt durch interaktive Begriffserklärungen wird nicht nur das allgemeine Textverständnis verbessert, sondern auch ein

zusätzlicher Lerneffekt für den Nutzer erzielt. Ebenso konnte nachgewiesen werden, dass eine Phonem-basierte Ausspracheanalyse technisch umsetzbar ist und Lernenden konkrete Hinweise zur Verbesserung ihrer Aussprache geben kann.

Gleichzeitig wurde aber an vielen Punkten ersichtlich, dass KI-Systeme klare Grenzen haben. Fehler beim Umwandeln in Leichte Sprache können bei der Textverarbeitung nicht geprüft werden. Ergebnisse können also immer Fehler oder Unsauberkeiten hinterlassen. Genauso hat auch die Ausspracheanalyse das Problem, dass das Feedback eher oberflächlich ist und die KI auch hier durch zahlreich auftretende Störfaktoren Fehler machen kann.

Dennoch stellt das Projekt „klarText - Barrierefreies Lernen“ einen praktikablen und skalierbaren Ansatz dar, um bestehende Sprachlernangebote sinnvoll zu ergänzen.

Insgesamt zeigt die Arbeit, dass KI einen relevanten Beitrag zur sprachlichen Integration und zum barrierefreien Zugang zu Informationen leisten kann, sofern sie verantwortungsvoll eingesetzt und kritisch reflektiert wird.

7. Literaturverzeichnis

- [1] BiB (Bundesinstitut für Bevölkerungsforschung), „Abwanderung aus Deutschland,“ 2026. [Online]. Available: <https://www.bib.bund.de/DE/Fakten/Fakt/M01-Zuzuge-Fortzuge-Deutschland-ab-1950.html>. [Zugriff am 25 Januar 2026].
- [2] Statistisches Bundesamt, „Bevölkerungsvorausberechnung für Deutschland,“ 2025. [Online]. Available: <https://service.destatis.de/bevoelkerungspyramide/>. [Zugriff am 16 Dezember 2025].
- [3] O. Struck, „Probleme der Arbeitsmarktintegration von Geflüchteten,“ *WSI-Mitteilungen*, p. 105–113, 2018.
- [4] G. Trauernicht, „Wie Kinder sprechen lernen,“ Juni 2002.
- [5] M. Richter, „Die Leichte Sprache,“ 25 März 2025. [Online]. Available: https://www.barrierefreiheit-dienstekonsolidierung.bund.de/Webs/PB/DE/barrierefreie_it/uebergreifende-anforderungen-web-und-app/leichte-sprache/leichte-sprache-node.html. [Zugriff am 22 Dezember 2025].
- [6] Netzwerk Leichte Sprache e.V., „Die Regeln für Leichte Sprache,“ 2022.
- [7] Bundesfachstelle Barrierefreiheit, „Neu: DIN SPEC Leichte Sprache veröffentlicht,“ 4 März 2025. [Online]. Available: <https://www.bundesfachstelle-barrierefreiheit.de/SharedDocs/Kurzmeldungen/DE/din-spec-leichte-sprache-veroeffentlicht>. [Zugriff am 12 Januar 2026].
- [8] DIN Media GmbH, „Empfehlungen für Deutsche Leichte Sprache,“ März 2025. [Online]. Available: <https://www.dinmedia.de/de/technische-regel/din-spec-33429/387728031>.
- [9] K. Carstensen, C. Ebert, C. Endriss, S. Jekat, R. Klabunde und H. Langer, „Eine Einführung,“ *Computerlinguistik und Sprachtechnologie*, Nr. 3, 2010.
- [10] Wikipedia Autoren, „International Phonetic Alphabet,“ 22 Januar 2026. [Online]. Available: https://en.wikipedia.org/wiki/International_Phonetic_Alphabet. [Zugriff am 23 Januar 2026].
- [11] R. Schäfer, „Was ist HTML?,“ 12 Juli 2022. [Online]. Available: <https://www.manitu.de/blog/20/Was-ist-HTML/>. [Zugriff am 2 Dezember 2025].

- [12] W3 Schools, „TypeScript Introduction,“ 2026. [Online]. Available: https://www.w3schools.com/typescript/typescript_intro.php. [Zugriff am 2 Dezember 2025].
- [13] Host Europe, „Angular: Das Frontend-Framework für moderne Webanwendungen,“ 13 August 2024. [Online]. Available: <https://www.hosteurope.de/blog/angular-das-frontend-framework-fuer-moderne-webanwendungen/>. [Zugriff am 2 Dezember 2025].
- [14] C. Staff, „KI-Programmiersprache: Was man im Jahr 2026 wissen sollte,“ 6 Dezember 2025. [Online]. Available: <https://www.coursera.org/de-DE/articles/ai-programming-languages>. [Zugriff am 27 Dezember 2025].
- [15] A. Chia, „Anaconda vs Python: Ihre Unterschiede erforschen,“ 11 Dezember 2024. [Online]. Available: <https://www.datacamp.com/de/blog/anaconda-vs-python-key-differences>. [Zugriff am 22 Dezember 2025].
- [16] D. Silva, „Lokale KI-Lösungen – einfach erklärt und sinnvoll genutzt!,“ 2 Januar 2026. [Online]. Available: <https://edih-saarland.de/de/lokale-ki-loesungen-einfach-erklaert-und-sinnvoll-genutzt-2/>. [Zugriff am 12 Januar 2026].
- [17] Movavi Content-Team, „Was ist eine WAV-Datei und wie wird sie verwendet?,“ 17 September 2025. [Online]. Available: <https://www.movavi.de/learning-portal/wav-file.html>. [Zugriff am 12 Januar 2026].
- [18] A. Dürkop, „Transkribieren mit Whisper,“ 5 Februar 2023. [Online]. Available: <https://axel-duerkop.de/post/2023-02-05-whisper-transcription/>. [Zugriff am 28 Dezember 2025].
- [19] C. Custer, „What is a UUID, and what is it used for?,“ 23 Juni 2023. [Online]. Available: <https://www.cockroachlabs.com/blog/what-is-a-uuid/>. [Zugriff am 5 Januar 2026].
- [20] M. McAuliffe, „Montreal Forced Aligner documentation,“ 2025. [Online]. Available: <https://montreal-forced-aligner.readthedocs.io/en/latest/>. [Zugriff am 18 Januar 2026].
- [21] M. McAuliffe, „Montreal-Forced-Aligner,“ 15 November 2025. [Online]. Available: <https://github.com/MontrealCorpusTools/Montreal-Forced-Aligner>. [Zugriff am 12 Januar 2026].

- [22] S. Sudhakara, „Goodness-of-Pronunciation,“ 20 Juli 2025. [Online]. Available: <https://github.com/sweekarsud/Goodness-of-Pronunciation>. [Zugriff am 12 Januar 2026].
- [23] S. Kanters, C. Cucchiarini und H. Strik, „The Goodness of Pronunciation Algorithm: a Detailed Performance Study,“ pp. 49-52, 2009.
- [24] M. Fräckiewicz, „TechStock,“ 16 Juli 2025. [Online]. Available: <https://ts2.tech/en/googles-gemini-ai-the-multimodal-supermodel-aiming-to-outshine-gpt-4-and-beyond/>. [Zugriff am 2026 Januar 12].
- [25] Stanford Report, „Die Macht der Sprache: Wie Worte Menschen und Kultur prägen,“ 22 August 2019. [Online]. Available: <https://news.stanford.edu/stories/2019/08/the-power-of-language-how-words-shape-people-culture>. [Zugriff am 24 Januar 2026].

8. Abbildungsverzeichnis

| | |
|---|----|
| Abbildung 1: Beispielhafter Aufruf des Montreal Forced Aligners | 9 |
| Abbildung 2: JSON-Ergebnis, Phonem & GOP Score | 10 |
| Abbildung 3: Leichte Sprache UI-Ausgabe | 13 |
| Abbildung 4: Feedback Ausgabe | 14 |