

# TECHNOLOGIEVERBESSERUNG DER AUTOMATISCHEN SPRACHERKENNUNG FÜR DIE OBERSORBISCHE SPRACHE

## Projektkurzbericht

Ivan Kraljevski  
Frank Duckhorn  
Constanze Tschöpe  
Matthias Wolff\*

Fraunhofer Institut für Keramische Technologien und Systeme IKTS,  
Maria-Reiche-Straße 2, 01109 Dresden  
*\*Brandenburgische Technische Universität Cottbus–Senftenberg, Cottbus*

Kunde: Stiftung für das sorbische Volk, Postplatz 2, 02625 Bautzen

Externe Freigabe

Vertraulichkeit: offen

Cottbus, 20.05.2025

---

Abteilungsleiter

---

Projektleiter



Management  
System  
ISO 9001:2015  
ISO 14001:2015  
[www.tuv.com](http://www.tuv.com)  
ID 1100005194



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung.....</b>	<b>3</b>
<b>2</b>	<b>AP 1: Technologische Verbesserung des klassischen Spracherkenners.....</b>	<b>3</b>
2.1	Beschreibung .....	3
2.2	Trainingsumgebung und -konfiguration .....	4
2.3	Erweiterte Sprachanwendung .....	4
2.3.1	Bereitgestellte Leistungen .....	5
<b>3</b>	<b>AP 2: Technologieverbesserung bei der Entwicklung des Sprachmodells und ihre Verwendung in Kombination mit KI.....</b>	<b>5</b>
3.1	Beschreibung .....	5
3.2	AP 2.1: Re-Implementierung und Optimierung der Eigennamenerkennung (engl.: Named Entity Recognition, NER) für die Wortklassendetektion in Texten .....	5
3.2.1	Eigennamenerkennung (NER).....	6
3.2.1.1	Textuelle Daten .....	6
3.2.1.2	Regelbasierter NER-Parser.....	6
3.2.1.3	SpaCy Pipeline.....	7
3.2.2	Bereitgestellte Leistungen .....	7
3.3	AP 2.2: Entwicklung eines Modells zur Ergebniskorrektur von anderen KI-basierten Spracherkennern .....	7
3.3.1	Open AI Whisper und Transkription des HSB-Korpus.....	8
3.3.2	Verfeinerung eines neuen OpenAI Whisper Modells.....	8
3.3.3	Fehleranalyse .....	8
3.3.4	Textkorrektur .....	8
3.3.4.1	Phonetisaurus-FST .....	9
3.3.4.2	SymSpell.....	9
3.3.5	Bereitgestellte Leistungen .....	10
3.3.5.1	OpenAI Whisper .....	10
3.3.5.2	Phonetisaurus.....	11
3.3.5.3	SymSpell.....	11
3.3.5.4	Dokumentation .....	11
<b>4</b>	<b>AP 3: Veröffentlichung der Sprachressourcen.....</b>	<b>11</b>
4.1	Beschreibung .....	11
4.2	Sprachdaten.....	11
4.3	Bereitgestellte Leistungen .....	12
4.3.1	Veröffentlichung des Korpus.....	13
4.3.2	Augmentation der Veröffentlichung.....	13
4.3.3	KALDI-Datensätze .....	13
4.3.4	Veröffentlichung des HSB-Korpus .....	13
4.3.5	Dokumentation.....	13
4.3.6	Veröffentlichungen .....	13

# 1 Einleitung

Dieses Projekt zielt auf eine Weiterentwicklung der Obersorbischen Spracherkennungstechnologie (Sprache-zu-Text, Speech-to-Text, STT). Verfügbare Sprachressourcen kamen sowohl bei der Entwicklung traditioneller STT-Systeme als auch bei der Untersuchung von adaptierten, vortrainierten Ende-zu-Ende-Modellen zum Einsatz.

In der entwickelten Methode kamen Teilwort- und Wortklassenmodellierungen zum Einsatz aufgrund der Datenknappheit für die sorbische Sprache. Das Wortklassenmodell basiert auf endlichen Transduktoren, welche sowohl bei einer offline Syntaxanalyse als auch in der Decodierung des STT-Systems verwendet werden kann. Die Sprachmodellierung wendet die Wortklassenanalyse entweder auf ganzen Wörtern des Sprachkorpus, Teilwörtern oder beidem gleichzeitig an. Zusätzlich nutzt die Eigennamenerkennung während der Nachbearbeitung der erkannten Transkriptionen die gleichen Wortklassendefinitionen.

Dieser Ansatz reduziert deutlich die Anzahl Wörter, die nicht im Vokabular des Modells enthalten sind, und ermöglicht eine bessere Anpassbarkeit der Spracherkennung für domänenspezifische Anwendungen. Das System wurde für die Transkription von Kirchenpredigten in Obersorbisch implementiert und anhand einer Aufzeichnung getestet. Das domänenspezifische System erreichte eine vergleichbare Leistung wie ein angepasstes „OpenAI Whisper“-Modell, weist eine hohe Ressourceneffizienz auf und liefert zudem semantische Markierungen.

## 2 AP 1: Technologische Verbesserung des klassischen Spracherkenners

### 2.1 Beschreibung

Ziel des Arbeitspakets ist die Verbesserung der Spracherkennung für die Obersorbische Sprache aus dem vorangegangenen Projekt. Die folgenden Punkte zur Verbesserung aus dem Strategiedokument (Kapitel 6.2.4) aus dem vorherigen Projekt wurden implementiert:

- a) Erleichterung des akustischen Trainings durch Implementierung einer Trainingsmethode, ohne die Notwendigkeit einer vollständigen phonetischen Annotation der Trainingsdaten.
- b) Unterstützung akustischer Modelle basierend auf Neuronalen Netzen (TDNN-Modelle). Damit verbundene Arbeiten: Erweiterung der Trainingsprozedur, Anpassung der Spracherkennungssoftware zum Einbinden der neuen Modelle.
- c) Gegebenenfalls Einbindung neuer Sprachdaten beim Training der Modelle.

Zu liefernde Leistungen:

- Trainingsumgebung und -konfiguration.
- Erweiterte Anwendung zur Spracherkennung und Hilfssoftware mit Gebrauchsanweisungen.
- Trainierte Modelle (Hidden-Markov-Modell für Triphone (HMM) und Time Delay Neural Network (TDNN)).

- Dokumentation.

## 2.2 Trainingsumgebung und -konfiguration

Der neueste Sprachkorpus HSB-Corpus (v1.0) mit 76.212 Aufnahmen (inklusive augmentierte Daten) wurden für das Training des Triphone- und TDNN-Modells herangezogen. Darin kommen 163 Sprecher vor und die Gesamtlänge der Aufzeichnungen beträgt 97:51:28 h. Die neu aufgezeichneten Daten sind ebenso im Korpus enthalten. Die neu gelieferten Modelle können in der Auswertung schlechter aussehen als frühere Modelle, weil in den Mengen „test“ und „dev“ zusätzlich die augmentierten Daten beinhalten.

Der Sprachkorpus ist zurzeit nicht öffentlich zugänglich.

Die existierenden Lexikon- und Sprachmodelle können mit neuen akustischen Modellen verwendet werden.

Eine Schritt-für-Schritt-Anleitung wurde in der Datei „README.md“ mitgeliefert.

## 2.3 Erweiterte Sprachanwendung

Fraunhofer IKTS entwickelt die proprietäre Software zur Sprach- und Signalerkennung (recIKTS). Die Software ist sowohl als eigenständige Anwendung als auch Programm-Bibliothek verfügbar und wurde in C und C++ verfasst. Sie ist kompatibel mit verschiedenen Architekturen und Betriebsumgebungen, wie zum Beispiel Win-32/64 oder Linux-i386/amd64/arm64, und ist optimiert zur Signalverarbeitung.

Die Software handhabt die gesamte Verarbeitungslinie, vom Audioinput über Merkmalsextraktion, I-Vektorberechnung, akustischen Modellberechnung bis zur Dekodierung. Fokus bei der Implementierung liegt auf Ressourceneffizienz.

Als Merkmalsextraktion unterstützt die Software klassische Mel-Frequenz-Cepstrum-Koeffizienten (MFCC), wie beispielsweise die Merkmalstransformation mittels lineare Diskriminanzanalyse (LDA), welche auch mit den Merkmalen des KALDI-Toolkits kompatibel sind. Alternativ bietet sie anpassbare Fourier-, Wavelet oder Cepstrumtransformationen für technische Signale an. Diese können mit konfigurierbaren Bandpassfiltern und Hauptkomponentenanalyse kombiniert werden. I-Vektorberechnungen für TDNN-Modelle werden ebenso unterstützt und können in Echtzeit eingesetzt werden.

Die akustischen Modelle werden außerhalb der Software trainiert und anschließend in die Anwendung importiert. Unterstützt werden TDNN-, HMM-Modelle von KALDI für Mono-, Bi- und Triphone und UASR-HMM-Monophone-Modelle.

Im Gegensatz zu KALDI bietet das Sprachmodell der IKTS-Software mehr Flexibilität. Sie unterstützt statistische Sprachmodelle im ARPA-Format, vordefinierte Kontextfreie Grammatiken (CFG) oder Automaten im OpenFST-Format. Zusätzlich kann das Sprachmodell Wortklassen integrieren aus vordefinierten Wortlisten, Grammatiken oder OpenFST-Automaten.

Ebenso lassen sich Teilwortmodelle einsetzen, um Teilwörter oder Morpheme bei dem Sprachmodell zu inkludieren, die während der Nachbearbeitung wieder zusammengesetzt werden.

Die Kernkomponenten des Spracherkenners nutzen einen für maximale Ressourceneffizienz optimierten Decoder basierend auf endlichen Transformatoren (engl.: Finite State Transducer, FST) und einer angepassten Tokenweitergabe. Der Decoder erlaubt eine

Detektion und eine Wiedergabe der Ergebnisse in Echtzeit für Aufzeichnungen variabler Länge. Dafür setzt die Software einen iterativen Backtracking-Algorithmus ein, um die Ergebnisse zu verbessern.

Neben der Spracherkennung an sich, kann die Software Sprechpausen detektieren. Für die Sprachpausenerkennung (engl.: Voice Activity Detection, VAD) kommen Gaußsche Mischmodelle (GMM) vom Open-Source-Projekt WebRTC. Ein entsprechender Automat handhabt den Auslösezeitpunkt, die minimale Aktivierungsstärke und die Deaktivierung.

### 2.3.1 Bereitgestellte Leistungen

Das gelieferte Paket zu AP 1 beinhaltet die Konfiguration, die Werkzeuge und trainierten Modelle (TDNN/HMM).

Die letzte Version der IKTS-Software (1.1.7) mit Beispielen zur Konfiguration sind ebenso inkludiert.

Zusätzlich werden dieser Abschnitt des Projektberichts und eine entsprechende Datei „README.md“ mitgeliefert.

AP 2: Technologieverbesserung...  
bei der Entwicklung des  
Sprachmodells und ihre  
Verwendung in Kombination mit  
KI  
-----

## 3 AP 2: Technologieverbesserung bei der Entwicklung des Sprachmodells und ihre Verwendung in Kombination mit KI

### 3.1 Beschreibung

Ziel des Arbeitspakets ist die Verbesserung der Sprachressourcen für das Training des Sprachmodells für Obersorbisch und die Steigerung der Genauigkeit bei der Spracherkennung. Die folgenden Punkte aus Kapitel 6.2.4 des Strategiedokuments aus dem vorherigen Projekt wurden implementiert.

### 3.2 AP 2.1: Re-Implementierung und Optimierung der Eigennamenerkennung (engl.: Named Entity Recognition, NER) für die Wortklassendetektion in Texten

Die folgenden Arbeitsschritte wurden durchgeführt:

- a) Automatische Annotation von Texten mit dem NER-Parser des Fraunhofer IKTS und eines Sprachmodells mit einer vordefinierten Liste von Entitäten (Personen, Orte, Daten, Zeiten, Organisationen, etc.)
- b) Qualitätskontrolle der Annotation
- c) Erstellung einer SpaCy-Spracherkennungspipeline (SpaCy: Open-Source-Python-Bibliothek zur Sprachverarbeitung) zur NER für Obersorbisch mit generierten Daten.
- d) Nutzung der Pipeline zur Markierung von Wortklassen für ein N-Gram-Sprachmodell für spezifische Domänen oder zur Nachbearbeitung automatisch generierter Transkripte.

Zu liefernde Leistungen:

- Pipeline für die Wortklassendetektion
- Dokumentation und Bericht der Ergebnisse.

Voraussetzungen (Kooperationsverpflichtungen des Kunden):

- Bereitstellung von Texten und Sprachdaten für die Entwicklung und Evaluation der Modelle
- Unterstützung bei der Identifikation von Fehlern und Qualitätskontrolle der Annotationen.

### 3.2.1 Eigennamenerkennung (NER)

Der NER-Parser wurde umgestaltet und optimiert. Die Wortklassendefinitionen, die für die STT ausgelegt waren, wurden vereinfacht und datengetrieben erweitert mit den Beispielen der „misa“-Textdaten.

#### 3.2.1.1 Textuelle Daten

Die Daten stammen aus verschiedenen öffentlichen Quellen und enthält religiöse Texte, wie zum Beispiel:

- „Church services“ - Transkription von aufgenommener Sprache
- „Wozjewjenja“ - geschriebene Ankündigungen im PDF-Format

Entwickelte Werkzeuge:

Das Script „misa2corp.sh“ erstellt die Trainings- (train.corp) und Testmengen (test.corp), wobei die Texte sowohl von Transkripten als auch Ankündigen stammen.

#### 3.2.1.2 Regelbasierter NER-Parser

Die Hauptaufgabe im AP 2.1 war die Optimierung und Anpassung des NER-Parsers (nlp/ner.py) zur Verbesserung seiner Geschwindigkeit und Genauigkeit.

Nennenswerte Veränderungen:

- Vorausladen von Wortklassendefinitionen,
- Schleifenbildung über Wortklassen und dann Eingabezeilen,
- Sortierung nach Länge und Entfernen von überlappenden Entitäten,
- Unterstützung von Zeitangaben, Prozenten und Währungen,
- Unterstützung von Vor-, Nachnamen und Ortsbezeichnungen
- Handhabung von Flexemen durch Abgleichen von Wortstämmen und
- Ausgabe der Ergebnisse als Text und JSON-Format, das für das Training der Python-SpaCy-Pipeline geeignet ist.

Das Script verwendet eine Reihe an Wortklassen („wordclass\_parser.txt“) mit vorberechneten FSTs:

{WÄHRUNG}	fst/currency.fst	*NEW*
{PROZENTE}	fst/percent.fst	*NEW*
{DATUM}	fst/DateUni.fst	*Aktualisiert: Datumsbereich*
{WOCHENTAG}	fst/weekdays.fst	
{UHRZEIT}	fst/ClockUni.fst	*Aktualisiert: vereinfacht*
{ORDINAL}	fst/ORDUni1-31.fst	
{CARDINAL}	fst/NUM1-10^6.fst	*Aktualisiert: von 0 bis 999*
{NAME}	fst/names.fst	*Aktualisiert: Namen*

{VORNAME} fst/surnames.fst \*Aktualisiert: Vornamen\*  
{ORT} fst/places.fst \*Aktualisiert: Orte\*

AP 2: Technologieverbesserung...  
bei der Entwicklung des  
Sprachmodells und ihre  
Verwendung in Kombination mit  
KI  
-----

Die Datei `README.md` enthält eine Schritt-für-Schritt-Anweisung und wurde als Teil der zu liefernden Ergebnisse übergeben.

Der annotierte Textkorpus kann bei der Erstellung von beliebigen anderen N-gram Sprachmodellen mit den vorhandenen Werkzeugen eingesetzt werden unabhängig von der Wort-/Teilwort-Tokenisierung. Die Wörter der Wortklassen werden nicht in Tokens übersetzt.

Das Debugging und die Qualitätsprüfung der Annotationen wurden anhand ausgewählter Sätze mit zielgruppenspezifischen FST-Grammatiken durchgeführt. Die Qualität wurde ebenso bei Experimenten der SpaCy-Pipeline bestätigt, wobei die annotierten Texte beim erfolgreichen Training und zur Evaluation eingesetzt wurden.

### 3.2.1.3 SpaCy-Pipeline

Zum Training und Testen der SpaCy-NER-Pipeline kamen jeweils die Dateien „`train_spacy.json`“ und „`test_spacy.json`“ zum Einsatz. Diese wurden dafür extra vom regelbasierten NER-Parser generiert oder manuell annotiert.

Die Evaluation der Testausgaben erfolgte mit den Metriken: Genauigkeit, Präzision, Recall und F-Score pro Token, Entität und Entitätstypen.

### 3.2.2 Bereitgestellte Leistungen

Vom AP 2.1 wurden die Software und Ressourcen bereitgestellt, die bei den einzelnen Zielstellungen zum Einsatz kamen (Scripte zur Erstellung und Konvertierung des Textkorpus, der NER-Parser, die SpaCy-Pipeline und Wortklassendefinitionen).

Die Dokumentation und eine Schritt-für-Schritt-Anleitung wurden in der entsprechenden Datei „`README.md`“ mitgeliefert.

## 3.3 AP 2.2: Entwicklung eines Modells zur Ergebniskorrektur von anderen KI-basierten Spracherkennern

Die folgenden Arbeitsschritte wurden durchgeführt:

- Anwendung von OpenAI Whisper auf alle Daten des HSB-Korpus
- Vergleich der Ergebnisse mit den Referenztranskripten.
- Identifikation falscher Wörter und Erstellung eines Datensatzes mit Wortpaaren (korrekt und falsch).
- Training eines KI-Modells (FST, neuronales Netz, etc.) für die Umwandlung falscher Wortklassen zu den Korrekten.
- Evaluation eines alternativen Ansatzes: Lerntransfer eines existierenden Sprachmodells (Transfer Learning).

Zu liefernde Leistungen:

- Ein erfolgreiches Modell zur Ergebniskorrektur.
- Dokumentation und Bericht der Ergebnisse.

Voraussetzungen (Kooperationsverpflichtungen des Kunden):

- Bereitstellung von Texten und Sprachdaten für die Entwicklung und Evaluation von Modellen.
- Unterstützung bei der Identifikation von Fehlern und Qualitätsprüfung von Annotationen.

### 3.3.1 Open AI Whisper und Transkription des HSB-Korpus

Die Transkription des HSB-Korpus erfolgte mit Hilfe eines verfeinerten, kleinen OpenAI Whisper Modells im GGML-Format. Die Ergebnisse wurden in eine CSV-Datei geschrieben.

Die bereitgestellte Datei „README.md“ enthält dafür eine Schritt-für-Schritt-Anleitung.

### 3.3.2 Verfeinerung eines neuen OpenAI Whisper Modells

Die Feinabstimmung des Modells auf den aktuellen HSB-Korpus erfolgte mittels eines „HuggingFace“-Moduls und der neusten Version des HSB-Korpus (v1.0).

### 3.3.3 Fehleranalyse

Für jeden Satz mit einem Schreibfehler wurden ein Paar bestehend aus der fehlerbehafteten und korrekten Wortsequenz festgehalten.

Die Transkripte ergaben die folgenden Wortfehlerraten (engl.: Word Error Rate, WER) und Zeichenfehlerraten (engl.: Character Error Rate, CER) auf dem aktuellen HSB-Korpus (v1.0) und dem unabhängigen „misa150“-Datensatz:

Datensatz	WER (%)	CER (%)
train_set_v3	1.73	0.30
test_set_v3	7.04	1.17
dev_set_v3	12.49	2.10
misa150_v3	23.27	4.40

Verglichen mit dem alten (Version 1) OpenAI Whisper HSB (Teile der Datensätze „test“ und „dev“ wurden beim Training verwendet):

Datensatz	WER (%)	CER (%)
test_set_v1	12.49	2.80
dev_set_v1	10.41	1.87
misa150_v1	33.17	6.14

### 3.3.4 Textkorrektur

Die transkribierten Sätze des gesamten HSB-Korpus wurden zusammen mit den korrekten Sätzen in einer CSV-Datei abgelegt. Augmentationen für zusätzliche Transkriptionsfehler wurden dem Datensatz „dataset\_v3.csv“ hinzugefügt und in Trainings- und Testdaten aufgeteilt. Daraus ergaben sich die folgenden WER/CER:

Datensatz	WER (%)	CER (%)
-----------	---------	---------

dataset_v3.csv	2.43	0.42
train_aug.csv	54.69	10.86
test_aug.csv	54.79	10.84

Die Aufteilung des Datensatzes „dataset\_v3.csv“ hat die folgenden WER/CER:

Datensatz	WER (%)	CER (%)
train_70K.csv	2.44	0.43
test_7K.csv	2.33	0.41

Die resultierenden, kombinierten Datensätze (train\_70K + train\_aug, test\_7k + test\_aug) besitzt die folgenden WER/CER:

Datensatz	WER (%)	CER (%)
train.csv	36.39	7.21
test.csv	43.56	8.61

### 3.3.4.1 Phonetisaurus-FST

Der Ordner „Text\_Correction/phonetisaurus“ enthält die Scripte für das Training und Testen vom Phonetisaurus-G2P-Modell (Graphem zu Phonem, G2P) zu Textkorrekturen. Das Modell erzielte die folgenden Ergebnisse auf dem Datensatz „test.csv“ bei nur falsch transkribierten (misspelled) oder nur korrekt transkribierten (corret) Sätzen mit und ohne Textkorrektur:

Datensatz	WER (%)		CER (%)	
	ohne	mit	ohne	mit
misspelled	29.35	18.97	11.08	4.77
correct	0.00	0.14	0.00	0.03

Das Phonetisaurus-Modell wurde nur mit Wortpaaren trainiert und kann deshalb keine Wörter berichtigen, die zusammengefügt oder getrennt wurden.

Auch wenn die meisten Wörter korrekt berichtigt wurden, ist das Modell fehleranfällig bei bisher unbekanntem Wörtern. Das Modell verändert nur selten richtig erkannte Wörter. Fraglich ist auch, inwiefern die zusätzliche Berechnungszeit der Korrekturen sich auf die Echtzeitanwendbarkeit auswirkt.

Daher ist dieser Ansatz nur teilweise für die Korrektur von Transkripten geeignet.

### 3.3.4.2 SymSpell

Das offizielle Repository der Software für Textkorrekturen von SymSpell ist:

<https://github.com/wolfgarbe/SymSpell>

und der Python-Wrapper:

<https://github.com/mammothb/sympellpy.git>

Für die optionale Neubewertung wird das Sprachmodell KenLM ARPA verwendet, das über einen Pfad zu den Binärdateien installiert werden.

Datensatz	WER (%)		CER (%)	
	ohne	mit	ohne	mit
test.csv	ohne	mit	ohne	mit
Gesamt	42.39	22.76	9.20	6.10
misspelled	54.34	29.18	11.79	7.81
correct	0.00	0.00	0.00	0.00

Mit Hilfe des zusätzlichen Sprachmodells (`gold.arpa`) zur Neubewertung der Worthyphese sind die Ergebnisse auf dem Testdatensatz:

Datensatz	WER (%)		CER (%)	
	ohne	mit	ohne	mit
test.csv	ohne	mit	ohne	mit
Gesamt	42.39	32.60	9.20	9.83
misspelled	54.34	37.18	11.79	11.06
correct	0.00	0.00	16.36	5.45

Es ist zu erkennen, dass das zusätzliche Sprachmodell zur Neubewertung schlechtere Ergebnisse liefert, weil insbesondere bereits korrekt transkribierte Wörter falsch verändert wurden. Entsprechend ist es wichtig möglichst viele Texte der jeweiligen Domäne zur Verfügung zu haben.

Die Verwendung von Unigramme und Bigramme führte zu besseren Ergebnissen, wobei die WER von 42,39 % auf 22,76 % reduziert wurde.

Wie bereits beim Phonetisaurus-Modell, falsch geschriebene Wörter mit mehr als zwei fehlerhaften Zeichen lassen sich nur schwer berichtigen aufgrund der Vielzahl von möglichen Wortkandidaten.

Der Unterschied bei der WER und CER in der obigen Tabelle ist darauf zurückzuführen, dass sie zunächst pro Satz gemittelt wurden und nicht direkt über den gesamten Datensatz.

Diese Methode lässt sich effizient berechnen selbst für Vokabulare mit mehreren Millionen Wortformen. Jedoch erzielt sie schlechtere Ergebnisse im Fall von mehreren falschen Zeichen pro Wort.

### 3.3.5 Bereitgestellte Lieferungen

Die folgende Software und Ressourcen wurden im Rahmen dieses Arbeitspakets bereitgestellt.

#### 3.3.5.1 OpenAI Whisper

Der Ordner „`OpenAI_whisper/`“ beinhaltet die Werkzeuge zur Datenvorbereitung, Training und Testen, das alte und neue verfeinerte kleine Whisper-Modell mit ihren quantisierten Versionen, die Skripte zur Fehleranalyse und die Ergebnisse der Transkripte in CSV-Dateien.

### 3.3.5.2 Phonetisaurus

Der Ordner „**phonetisaurus/**“ enthält die Werkzeuge zur Datenvorbereitung, Training und Evaluation zusammen mit den Aufzeichnungen der Ergebnisse.

### 3.3.5.3 SymSpell

Im Ordner „**symspell/**“ sind die Werkzeuge zur Unigramm/Bigramm-Schätzung, Evaluationscripte und die Ergebnisse der Textkorrektur abgelegt.

### 3.3.5.4 Dokumentation

Dieser Teil der Projektberichts und die korrespondierende „**README.md**“ werden ebenso bereitgestellt.

## 4 AP 3: Veröffentlichung der Sprachressourcen

### 4.1 Beschreibung

Die Veröffentlichung der Sprachdaten ermöglicht andere Gruppen die Arbeit an einer obersorbischen Spracherkennung.

Außerdem ist die Voraussetzung für die Erstellung wissenschaftlicher Veröffentlichungen in diesem Bereich.

Dieses AP besteht aus den folgenden Teilaufgaben:

- a) Identifikation von Sprachaufnahmen, die veröffentlicht werden können.
- b) Erstellung von Metadaten, Lizenzinformationen und Dokumentation (analog zu LibriSpeech).
- c) Vorbereitung eines technischen Papers für die Beschreibung des Korpus und der Übermittlung an ESSV25.
- d) Veröffentlichung der Daten auf GitHub oder als Archiv mit Link zum Herunterladen.

Zu liefernde Leistungen:

- Für die Veröffentlichung bearbeitete Sprach- und Textdaten
- Gemeinsames Paper zur Veröffentlichung

Voraussetzungen (Kooperationsverpflichtungen des Kunden):

- Unterstützung bei rechtlichen Angelegenheiten und Einholen der Einwilligungen der Sprecher

### 4.2 Sprachdaten

Der Sprachkorpus (HSB-Korpus) ist in mehrere Repositorien unterteilt. Sie repräsentieren spezifische Domänen: Pilot-Projekt (HSB), zehn Filme (SCF), zehn Studioaufnahmen (SCM), männliche (SCK) und weibliche (SCW) Sprecher für Text-zu-Sprache-Systeme (engl.: Text-to-Speech, TTS) und die validierte Teilmenge des Common Voice HSB-Datensatzes v5.1 (CV). Die Repositorien werden mit jeder Veröffentlichung aktualisiert. Die Aufzeichnungen liegen im 16-bit PCM WAV-Format mit 16 kHz vor. Ein extra Ordner enthält die Transkripte mit den entsprechenden Dateien. Auf jeder Zeile ist immer

nur ein Wort in Großbuchstaben geschrieben. Die Textdateien sind UTF-8-encodiert und frei von Sonderzeichen.

Jedes Audio-Transkript-Paar besitzt einen allgemeinen, einzigartigen Identifikator (engl.: Universally Unique Identifier, UUID) während der Zusammenstellung der Veröffentlichung. Deswegen verwendet jede neue Version des Korpus aktualisierte Dateinamen für Aufzeichnungen und Transkripte. Dieser Vorgang hilft dabei die Dateireihenfolge zu randomisieren und somit die Anonymität der Sprecher in Filmen und Dokumentationen sicherzustellen.

Der Korpus enthält Augmentationen, u. a. mit Gaußschen und realen Hintergrundgeräuschen, Zeitverzerrungen, -verschiebungen und simulierten Echos von großen Innenräumen, wodurch die Datenmenge effektiv verdoppelt wird.

Die folgende Tabelle zeigt wesentliche Statistiken des Korpus darunter der Wortschatzreichtum, gemessen am Typ-Token-Verhältnis (engl.: Type-Token Ratio, TTR). Dieses Verhältnis vergleicht die Anzahl einzigartiger Wörter (Typ) mit der Gesamtanzahl an Wörtern (Token) im Text.

	Kombiniert	HSB	SCF	SCM	CV	SCK	SCW
#Sprecher	163	31	106	23	16	1	1
#Aufnahmen	38106	6313	14439	12473	1350	2442	1089
Dauer	48:56:23	11:34:40	08:06:25	21:46:20	02:27:58	02:48:27	02:12:32
Ø Dauer (s)	4.62	6.6	2.02	6.28	6.58	4.14	7.3
Ø Anzahl Wörter	7.74	7.29	4.6	11.07	9.57	7.97	11.18
Typ-Token-Verhältnis	0.13	0.08	0.11	0.21	0.43	0.28	0.47

Der daraus resultierende Sprachkorpus beinhaltet die augmentierten Aufzeichnungen mit über 70.000 Sätzen von 163 Sprechern und insgesamt mehr als 97 h Audio. Die Berechnung des Korpus ist konfigurierbar und Sprecher oder andere Teilmengen können von der Veröffentlichung exkludiert werden.

### 4.3 Bereitgestellte Lieferungen

Der obersorbische Sprachkorpus besteht aus verschiedene Sprachkorpen aus unterschiedlichen Domänen: HSB aus einer Machbarkeitsstudie (Smart Lamp, Sprachsteuerung einer Lampe), Filme (SCF), Aufzeichnungen (SCM) und die validierte Teilmenge des Common Voice HSB Datensatzes v5.1 (CV).

Sie sind als Teilmodule im Repository und bei jeder Veröffentlichung aufgerufen.

<https://github.com/kogmatd/hsbcorpus.git>

Die Version der Veröffentlichung ist in der Datei „VERSION“ aufgeführt.

Die Lizenz der Veröffentlichung ist in der Datei „LICENSE“ enthalten.

Eine Schritt-für-Schritt-Anweisung wurde mit der Datei „README.md“ mitgeliefert.

#### 4.3.1 Veröffentlichung des Korpus

Ein automatisches Script holt die Daten vom Repository und dessen Teilmodule, hebt ggf. Änderungen auf und ruft Korpus-spezifische Vorverarbeitungsscripte auf, um die Formate von Audio und Transkripte und Namenskonventionen zu vereinheitlichen.

#### 4.3.2 Augmentation der Veröffentlichung

Die Veröffentlichungen können optional mit augmentierten Audiodateien erweitert werden, indem Gaußsches Rauschen, Zeitverzerrungen, -verschiebungen, Hintergrundgeräusche oder andere Effekte von Raumsimulationen hinzugefügt werden.

Durch das hinzufügen der Ordner mit dem Suffix „\_AUG“ kann die Datenmenge der Veröffentlichung effektiv verdoppelt werden. In diesem Fall muss die Dateiliste und Metadaten neu erzeugt werden.

#### 4.3.3 KALDI Datensätze

KALDI-kompatiblen Daten wurden auf einer lokalen Workstation mit absoluten Pfaden erstellt. Sie können mit einer entsprechenden KALDI-Verarbeitung weiterverwendet werden. Die KALDI-Daten enthalten absolute Pfade und sollten nicht Teil der öffentlichen Freigabe sein.

#### 4.3.4 Veröffentlichung des HSB-Korpus

Der Inhalt des Ordners „release\_\$VERSION/“, wo die Version der Veröffentlichung manuell in der Datei „VERSION“ gesetzt werden kann, muss ohne die „.log“-Dateien manuell in einer ZIP(-TAR)-Datei archiviert werden.

Das komprimierte Archiv muss als offizieller Release im öffentlichen Repository bereitgestellt.

#### 4.3.5 Dokumentation

Dieser Teil der Projektberichts und die korrespondierende „README.md“ werden ebenso bereitgestellt.

#### 4.3.6 Veröffentlichung

Kraljevski I., Duckhorn F., Sobe D., Tschöpe C., Wolff M., "Preserving Language Heritage Through Speech Technology: The Case of Upper Sorbian", International Conference on Speech and Computer, 2024, Springer Nature Switzerland Cham., [http://dx.doi.org/10.1007/978-3-031-77961-9\\_1](http://dx.doi.org/10.1007/978-3-031-77961-9_1)

Kraljevski I., Duckhorn F., Sobe D., Tschöpe C., Wolff M., "Speech-To-Text in Upper Sorbian: Current State", ESSV 2025, Halle/Salle, 3-5 March 2025.