Fraunhofer IKTS

FRAUNHOFER-INSTITUT FÜR KERAMISCHE TECHNOLOGIEN UND SYSTEME IKTS

VORBEREITUNG DER SPRACHERKENNUNG FÜR DAS OBERSORBISCHE FÜR EINE DIKTIERFUNKTION

Projektbericht

Ivan Kraljevski
Frank Duckhorn
Isidor Konrad Meier*
Johannes Ferdinand Joachim Kuhn*
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

Cottbus, 16.05.2023

Externe Freigabe

Inhalt

Einleitung

Dieses Projekt baut auf den Ergebnissen von zwei vorangegangen Projekten auf. Ziel des Projektes ist, die Erweiterung bestehender und Entwicklung neuer Mechanismen zur Handhabung großer Vokabulare für die obersorbische Spracherkennung.

Aktuell beschränkt sich die Erkennung auf einzelne, spezifische Domänen. In Zukunft können diese zusammengeführt werden, um eine domänenunabhängige Anwendung zu erstellen.

Für die Erkennung von stark flektierten Sprachen, wie zum Beispiel Obersorbisch, stellt die Sprachmodellierung eine große Herausforderung dar, weil viele Wörter und ihre Beugungsformen berücksichtigt werden müssen. Die Größe des Vokabulars und die Qualität des Sprachmodells beeinflussen direkt die Performanz der Spracherkennung.

Im Idealfall würde die Erstellung des Sprachmodells mit möglichst vielen domänenspezifischen gesprochenen Sätze und Texten erfolgen. Gleichzeitig sollte das genutzte Vokabular so klein wie möglich sein.

1 Bisherige Arbeiten

1.1 HSB-I: Machbarkeitsstudie für die automatische Spracherkennung von Obersorbisch

In einer ersten Machbarkeitsstudie (interne Bezeichnung: HSB-I) wurden Möglichkeiten untersucht, erlernte Kenntnisse für die Erkennung einer Sprache auf andere zu übertragen (engl. Transfer Learning). Bei bedrohten Sprachen, wie zum Beispiel Obersorbisch (ISO Sprachcode: HSB), existieren in der Regel nur wenige Ressourcen für das Erstellen von Sprachmodellen. Daher wurden zusätzlich Datensätze von verbreiteteren Sprachen (bspw. Deutsch) genutzt.

Für die Demonstration der Machbarkeit entstand ein Demonstrator zur automatischen Spracherkennung (engl. Automatic Speech Recognition, kurz: ASR) zur Steuerung von Lampen (Smart Lamp).

Die Hauptaufgaben definierten sich durch die folgenden Punkte:

- Definition von zu erkennenden Graphemen und Phonemen basierend auf den bereitgestellten Quellen des Projektpartners (Stiftung für das sorbische Volk)
- Definition einer einfachen Sprachanwendung (Demonstrator zur sprachgesteuerten smarten Lampe)
- Vorbereitung und Organisation der Datensammlung von Sprachaufnahmen zum Entwickeln und Testen der Sprachanwendung
- Umsetzung der Sprachanwendung:
    - Aufbereitung des Sprachkorpus
    - Phonemerkennung zur Evaluation des akustischen Modells
    - Optimierung der zu erkennenden Phoneme (Phoneminventar) und die Graphem-Phonem-Abbildung
    - Sprachmodellierung durch Grammatiken mit endlichen gewichteten Automaten (Finite-State Grammars)
    - Adaption eines einfachen akustischen Modells (Deutsch) für Obersorbisch mittels der gesammelten Daten
    - Evaluierung des Modells durch Testdaten

Während der Machbarkeitsstudie konnte die Grundlage zur Entwicklung von anspruchsvolleren Sprachanwendungen für Obersorbisch geschaffen werden: Phoneminventar, Sprachkorpus (ca. 11 Stunden), einfaches akustisches Modell (monophon) und Sprachmodell (Grammatik).
Die entwickelten Verfahren können außerdem zum Einsatz kommen, um weitere Daten zu sammeln und Sprachanwendungen für Niedersorbisch durch die Neudefinition der Graphem-Phonem-Abbildung zu entwickeln.

1.2 HSB-II: Verbesserung der Spracherkennung von Obersorbisch

Das zweite Projekt (interne Bezeichnung: HSB-II) zielte auf die akustische Modellierung ab basierend auf den gesammelten Daten mit einem nativen Phoneminventar, anstatt auf das Inventar einer anderen Sprache zurückzugreifen:

- Akustische Modellierung:
  - Aussprachemodellierung
  - Angleichen des Sprachkorpus mittels eines vortrainierten Modells
  - Training neuer, nativer akustische Modelle für Obersorbisch (dLabPro/UASR)
  - Sprecherabhängige Adaption
  - Evaluation von sprecherabhängigen und -unabhängigen akustischen Modellen
- Lexikonmodellierung:
  - Statistische, überwachte Graphem-zu-Phonem-Modellierung
  - Open-Source-Toolkits mit verfügbaren Trainingsdaten aus Wort-Phonem-Abbildungen
- Sprachmodellierung:
  - Umsetzung kontextfreier Grammatiken (Finite-State Grammars) als Transduktoren mit endlichen Zuständen (engl. Finite-State Transducers)
  - Grammatiken für die Zuordnung von Wortklassen (Zeit, Datum, Zahlen)
  - Statistischen Sprachmodell mit Wortklassen
- Best-Practice-Richtlinien für die Erstellung von Sprachkorpora

Des Weiteren ließen sich die Funktionalitäten und Prozeduren zur akustischen, Lexikon- und Sprachmodellierung im Projekt verbessern, womit domänenspezifische Anwendungen mit kleinem bis mittlerem Vokabular erstellt werden können.

Die angewandten Technologien sind immer noch sehr abhängig von der verfügbaren Sprach- und Textmenge. Beispielsweise. ist mit dem vorhandenen obersorbischen Sprachkorpus nur das Training von monophonen akustischen Modellen realisierbar. Triphon-Modelle würden schlechter abschneiden. Der Einsatz von vortrainierten Triphon-Modellen oder hybriden tiefe neuronale Netze (engl. Deep Neural Networks, kurz: DNN) mit verborgenen Markovmodellen (engl. Hidden-Markov-Models, kurz: HMM) von anderen Sprachen würden Fehler bei der Triphon-Erkennung in den adaptierten Daten hervorrufen.

Der Textdatensatz ist unzureichend für das Training von statistischen Sprachmodellen mit einem großen Vokabular. Deshalb eignen sich besonders kontextfreie Grammatiken in Kombination mit Wortklassen für die Sprachanwendungen wie zum Beispiel sprachkontrollierte persönliche Assistenten oder Smart-Home-Geräte für einzelne oder mehrere Sprecher (angepasste akustische Modelle).

Außerdem lassen sich kleine domänenspezifische Texte mit begrenztem Vokabular erfolgreich zur statistischen Sprachmodellierung mit Wortklassen einsetzen. Damit sind Sprachanwendungen mit flexiblen Ausdrücken möglich. Und selbst wenn das Gesprochene nicht komplett korrekt erkannt wird, könnte die Bedeutung (Semantik) robust abgeleitet werden.

1.3 HSB-III: Motivation

Im aktuellen Projekt wird die mögliche Entwicklung eines domänenunabhängigen Diktiersystem für Obersorbisch durch die Erweiterung und Verbesserung der folgenden Technologien untersucht:

- Triphon-basierte akustische Modelle und
- Statistische Sprachmodelle, welche auf Wortteilen (Silben oder Morpheme) basieren.

2 AP 1: Verbesserung der akustischen Modelle

2.1 Übersicht

Der Wechsel von monophon- auf triphon-basierte akustische Modelle erfordert die Sammlung von weiteren transkribierten Daten von ungezwungener Sprache aus verschiedenen Domänen.

2.2 Zielstellungen

Ziel des Arbeitspaketes ist das Training von akustischen Modellen auf größeren Audiokorpora, erweitert durch Datenaugmentationen:

- Training und Evaluation von neuen monophon- und triphon-basierten Modellen

- Tool zur Umwandlung von Features und Vorbereitungen von Daten für das Training mit dem Open-Source-Toolkit KALDI.

2.3 Ergebnisse

2.3.1 Sprachkorpus

Für die triphon-basierte akustische Modellierung kamen Sprachdaten aus drei Korpora aus verschiedenen Quellen zum Einsatz: Common Voice Projekt hsb-dataset (v5.1), gesammelte Daten aus der ersten Machbarkeitsstudie (HSB-I), und neue Audiodaten aus diesem Projekt (Speech Corpus Film - SCF).

Korpus	*Anzahl Sprecher (IDs)*	*Anzahl Sätze*	*Signallänge (HH:MM:SS)*
Common Voice	**17 versch. Sprecher**	**1.359**	**02:28:54**
Machbarkeitsstudie (HSB-I)	**31 versch. Sprecher**	**6.320**	**11:34:54**
HSB-01	11	2.094	3:33:50
HSB-02	11	2.170	3:22:30
HSB-03	11	2.056	4:38:34
Speech Corpus Film (HSB-III)	**58 versch. Sprecher**	**7.615**	**03:56:14**
speech_corpus_film_9_a_pol	4	219	00:08:29
speech_corpus_film_gilles	17	1.879	00:53:12
speech_corpus_film_karla_a_katrina	20	1.816	00:40:15
speech_corpus_film_mpz_insekten	1	457	00:28:08
speech_corpus_film_mpz_reise	1	494	00:26:38
speech_corpus_film_mpz_wjedro	1	446	00:31:25
speech_corpus_film_peeweeje	21	2.304	00:48:07
Gesamt	**106 versch. Sprecher**	**15.294**	**18:00:02**

Abb. 1 Anzahl von Sprechern, Sätzen und Signallänge von den verschiedenen Sprachkorpora.

2.3.2 Audioaugmentation

Störungsfreie Sprachaufnahmen können durch das Einfügen von Hintergrundgeräusche verschiedener Stärken augmentiert werden. Dadurch lässt sich der verfügbare Sprachdatensatz effektiv verdoppeln.

Die Augmentation bietet eine große, diverse Datenmenge, welche für die Erstellung von robusten Triphonmodellen notwendig ist.

Lediglich die Trainingsdaten für das akustische Modell wurden verändert. Der Common Voice Datensatz blieb unverändert, weil er zur Evaluation des Modells zum Einsatz kam.

Nach Augmentation umfassten die Sprachaufnahmen insgesamt `33:31:10` Stunden und `29.229` gesprochene Sätze.

2.3.3 Training des akustischen Models

Das Training des monophon- und triphon-basierten Modells erfolgte mit dem Open-Source-ASR-Toolkit KALDI. Zu Beginn mussten der Sprachkorpus, die Audiodaten und die Transliterationen in Konfigurationsdateien für KALDI umgewandelt werden.

2.3.4 Datensätze

Der Sprachkorpus wurde in drei Teile für Training (train), Evaluation (test) und Entwicklung (dev) aufgeteilt. Dabei wurde sichergestellt, dass kein Sprecher in mehreren Datensätzen gleichzeitig vorkommt.

- `train` mit 24.052 Sätzen, augmentierte Daten eingeschlossen,
- `test` mit 3.760 Sätzen, ähnliche Aufnahmebedingungen (Augmentationen inbegriffen),
- `dev` mit 1.350 Sätzen, Common Voice Datensatz, repräsentiert Anwendungsfälle.

2.3.5 KALDI

KALDI ist ein modernes Open-Source-Framework zur automatische Spracherkennung. Es gilt als eines der beliebtesten Toolkits mit einer umfangreichen Unterstützung und Sammlung von vortrainierten Modellen.

Es wurde in C++ implementiert, unter der Apache Lizenz v2.0 lizensiert und für den Einsatz bei der Spracherkennungsforschung konzipiert.

In diesem Projekt wurden damit verschiedene triphon-basierte akustische Modelle trainiert. Die Modelle lassen sich in das Format für `dLabPro/UASR` umwandeln und vom Fraunhofer IKTS zur Spracherkennung einsetzen.

2.3.6 Evaluation des akustischen Modells

Die Evaluation der Erkennungsleistung der akustischen Modelle wurde direkt nach ihrem Training durchgeführt.

Das Lexikon und die Sprachmodelle wurden aus den Transliterationen der Datensätze `training, dev,` and `test` erstellt. Das Sprachmodell ist ein statistisches 3-Gramm Modell, das durch das IRSTLM Toolkit erstellt wurde. Die Ergebnisse zeigen deswegen den optimalen Fall, bei dem die Testsätze im Sprachmodell vorkommen.

Die Kombination der Transliterationen ergibt einen Korpus mit verschiedenen Domänen (Alltägliches, Lampensteuerung, Untertitel für Filme und Dokumentationen). Das ist nicht optimal, aber durch den Vergleich sind Rückschlüsse auf die Modellqualität möglich.

Model	Test	Dev
monophone	12,52 [11,62,13,42]	10,40 [9,07,11,72]
hsb_500_20000_tri1	10,83 [10,04,11,63]	6,91 [6,10, 7,73]
hsb_1000_40000_tri2	10,14 [9,38, 10,90]	7,94 [7,08, 8,80]

Abb. 2 Wortfehlerrate (in %) mit 95% Konfidenzintervallen (engl. Confidence Intervals, kurz: CI).

Aus der Wortfehlerrate (engl. Word Error Rate, kur: WER) lässt sich erkennen, dass beide triphon-basierte Modelle eine signifikant bessere Leistung bieten, verglichen mit dem monophon-basierten Modell. Der Unterschiede zwischen der `tri1` und `tri2` Version sind nicht signifikant. Man kann also vom kleineren und schnelleren Modell bei der späteren Anwendung profitieren. Deshalb benutzen wir für die weitere Entwicklung und Evaluation das akustische Modell `tri1`.

2.4 Zusammenfassung

Ein neues, robusteres und weiterentwickeltes (triphon-basiert) akustische Modell wurde mit Hilfe des großen Sprachkorpus trainiert, der auch neue und augmentierte Aufnahmen umfasste.

3 AP 2: Verbesserung des Sprachmodells

3.1 Übersicht

Die Entwicklung einer kontinuierliche Spracherkennung für Obersorbisch mit einem umfangreichen Vokabular (engl. Large Vocabulary Continuous Speech Recognition, kurz: LVCSR) stellt eine große Herausforderung dar, u.a. durch die Struktur der Sprache und die unzureichende Verfügbarkeit von elektronischen Sprachressourcen.
Das LVCSR-System, das frei formulierte Sprache verlässlich erkennen kann, muss mit solchen Daten trainiert werden. Weil keine Transkripte für alltägliche Sprache existieren und diese in der Regel auch nicht komplett mit dem Gesprochenen übereinstimmen, muss zur Erstellung des Sprachmodells auf andere Quellen zurückgegriffen werden.

Daher wurden Texte aus sehr unterschiedlichen Domänen gesammelt, bspw. von Büchern, Filmuntertiteln, Zeitungen und Protokollen.
Folglich besteht das Vokabular aus weniger alltäglichen Wörtern und ihr Kontext lässt sich nur schlecht statistisch modellieren.

Idealerweise würde die zu erkennende Sprechweise und Formulierungen auch in den Trainingsdaten für das akustische Model, Lexikon und Sprachmodell vorkommen.

Ein weiteres Problem stellt der Umfang des Zielvokabulars dar. Insbesondere bei flektierten Sprachen kann ein Lexikon aus mehreren Millionen Wortformen bestehen. Dadurch kann der Anteil von Wörtern, die nicht vom Vokabular erfasst werden, (engl. Out Of Vocabulary rate, kurz: OOV) hoch und die Spracherkennung unverlässlich sein.

Eine mögliche Lösung ist die Verwendung von Wortteilen zur Modellierung. Hierfür unterteilt man einzelne Wörter in kleinere Abschnitte (Tokens), wodurch sich das Vokabular verkleinert. Im Extremfall werden nur Phoneme (oder Grapheme) als Grundeinheiten verwendet. Der Extremfall ist hier allerdings unpraktikabel, weil sie häufig akustisch verwechselt werden und eine korrekte Abfolge nur mit einem deutlich größeren Kontext bestimmbar ist.

Es existieren viele Ansätze zur Teilwortbildung für die Verarbeitung natürlicher Sprache (engl. Natural Language Processing, kurz: NLP) und zum Trainieren von großen Sprachmodellen (engl. Large Language Models, LLMs). In diesem Arbeitspaket kam das Verfahren Byte Pair Encoding (BPE) und der Morfessor- Algorithmus zum Einsatz.

3.2 Zielstellungen

Dieses Arbeitspaket zielt hauptsächlich auf die Verbesserung der Sprachmodellierung mit den bereitgestellten Ressourcen ab. Die unterschiedlichen Daten mussten organisiert, importiert und in passende Formate konvertiert werden. Zusätzlich erfolgten eine Vorverarbeitung und Normalisierung, so dass die Daten in den kommenden Schritten verarbeitet werden konnten.

Die Normalisierung bestand aus den folgenden Punkten:

- Import, Vorverarbeitung und Vereinheitlichung der verschiedenen Textformate (bspw. doc, pdf, html, xml),
- Normalisierung von Abkürzungen und Ersetzen mit Tags oder gesprochenen Wörtern, Enterfernen von redundanten Punktationen und Sonderzeichen,
- Erkennung von Eigennamen (engl. Named Entities Recognition, kurz: NER) aus vordefinierten Wortklassen (Namen, Zeit, Datum, Zahlen, Orte), Ersetzen von Wortklassen im normalisierten Korpus, Erstellung des Grammatikmodells,
- Segmentierung von Wörtern zu Tokens (morphologische Einheiten) mit Hilfe von automatischen Parsern, Überprüfen der Segmentierung basierend auf Punktation von Wörtern und Tokens,
- Generierung des Vokabulars und des Lexikons für die Teilwörter.

Für die Normalisierung wurde ein Python Skript erstellt und an die gegebenen Daten angepasst. Anschließend wurde ein Sprachmodell basierend auf den Tokens trainiert und bezüglich seiner Perplexität evaluiert.

3.3 Ergebnisse

3.3.1 Vorverarbeitung und Normalisierung der Texte

Normalisierung von Texten in Format mit einem Satz pro Zeile, durchgehender Großschreibung und Entfernung von Satz- und Sonderzeichen:

Zunächst werden die gelieferten Textdaten im UTF-8-Format eingelesen. Hierbei werden überflüssige Leerzeichen und Absätze entfernt. Anschließend wird der Text nach Abkürzungen durchsucht, um diese für die kommenden Schritte zu markieren, damit dort kein Satzende detektiert wird. Nun werden Sonderzeichen gefiltert, die z.B. häufig für Aufzählungen genutzt werden, sowie alle Arten von Klammern.

Anschließend werden die Sätze in erster Näherung durch bekannte Satztrennzeichen erkannt und in eine eigene Zeile geschrieben. Als letzten Schritt wird die Markierung der Abkürzungen wieder aufgehoben und die Texte für die weitere Verarbeitung in einem vom Benutzer definierten Ordner abgespeichert.

Sollten die Texte nicht wie gefordert als Textdatei im UTF-8-Format vorgelegen haben, so wurden diese in solche umgewandelt. Falls es sich um Word-Dateien handelte, wurde ein VBA-Code geschrieben, der die Word-Datei in einem Ordner automatisch in eine Text-Datei im UTF-8-Format ohne BOM umwandelt. Waren die Dateien im tei-Format, wurde eine Anleitung geschrieben, wie diese mit der Umwandlungswebseite in Word-Dateien umgewandelt werden können.

Trotzdem ist die Trennung der Sätze noch übersensitiv, dass heißt, es wird oft ein Satz in mehrere Zeilen aufgetrennt. Gründe dafür sind u.a. a) hart-gecodete Zeilenumbrüche und b) Punkte in Abkürzungen (wie 'na př.'), die als Satzende interpretiert werden. Daher haben wir für den nächsten Schritt ein Programm geschrieben, das Trennungen rückgängig gemacht, wenn 1.) hinter der Trennung ein Kleinbuchstabe folgt und 2.) vor der Trennung entweder a) kein Satzzeichen wie '.', '?', '!', ';' oder '...' steht oder b) Eine Abkürzung (wie na př.) steht. Dafür haben wir eine Liste von gängigen Abkürzungen die Punkte enthalten verwendet.

Nach den Wiedervereinigungen wurden alle Sätze den dobry_sady (gute Sätze) oder snano_dobry_sady (vielleicht gute) zugeteilt, oder ganz verworfen. 'Dobry_sady' müssen auf '.', '?', '!', ';' oder '...' enden und dürfen keine deutschen oder niedersorbischen Buchstaben wie 'ä', 'ö', 'ü', 'ŕ', 'ś' oder 'ź' (außer in 'dź') enthalten. 'Sna-

no_dobry_sady' dürfen deutsche oder niedersorbische Buchstaben enthalten. Sie müssen nicht auf ein Satzzeichen enden, sofern sie mehr als 3 Wörter haben und die folgende Zeile mit einem Großbuchstaben beginnt. 'Dobry_sady' werden nicht zusätzlich als 'snano_dobry_sady' aufgeführt.
Die 'snano_dobry_sady' dienen dem manuellen Erkennen von Mängeln in der Normalisierung. Die verwendeten Einstellungen wurden durch Ausprobieren so gewählt, dass sie nur wenige 'snano_dobry_sady' entstehen, die meistens auf größere Mängel hinweisen.

Anschließend können Mängel manuell behoben werden, Details sind im Workflow zu finden.
Danach wird abschließend normalisiert. Sofern keine manuellen Eingriffe erfolgt sind, würden dann nur noch 'dobry_sady' übrigbleiben, die auf ein Satzzeichen enden und mit einem Großbuchstaben oder einer 4-stellige Zahl anfangen. (4-stellige Zahlen werden als Jahreszahlen angenommen. Sonstige Zahlen stehen im Verdacht, Aufzählungspunkte oder (Bibel-)verse o.ä. zu sein.) Die verbleibenden Sätze liegen dann in Großschreibung ohne Satzzeichen vor.

3.3.2 Wortklassenmodellierung

Die Wortklassenmodellierung umfasst die Erkennung von benannten Einheiten aus vordefinierten Wortklassen (Namen, Zeit, Datum, Zahlen, Ortsangaben) und die Ersetzung der Wortklassen im Korpus und Erstellung von Grammatikmodellen für jede Wortklasse:
Für die Erkennung benannter Einheiten wurden einerseits die FSTs aus dem letzten Projekt HSB-II und andererseits Listen von Orten und Namen verwendet.
Wir haben ein Programm geschrieben, das nacheinander Datums-, Uhrzeits-, Wochentags-, Namens-, Orts- und schließlich Zahlen-Ausdrücke im Text durch Token ersetzt.
Das Format eines Tokens des Worts X ist:

- Bei Daten: `<DATE:DD-MM>`
- Bei Uhrzeiten: `<TIME:HH:MM>`
- Bei Wochentagen: `<WDAY:X>`
- Bei Namen: `<names:X>`
- Bei Orten: `<places:X>`
- Bei Ordinalzahlen: `<ONUM:x>`
- Bei Kardinalzahlen: `<CNUM:x>`

Dabei ist x die digitale (arabische) Schreibweise des Zahlwortes X. (X='jedyn' --> x=1)
Im Falle der FSTs sucht das Programm nach Erzeugnissen des Transducers, bei Listen dagegen nur nach Einträgen in der Liste.
Die Reihenfolge wurde absichtlich so gewählt, damit z.B. in Uhrzeiten nicht die enthaltenen Zahlen erkannt werden, ehe die Uhrzeit als ganze erkannt wird.
Die Erkennung von Namen und Orten ist noch nicht vollständig. Das könnte durch Bereitstellung vollständigerer bzw. exakterer Listen behoben werden.
Die Liste der Namen ist unvollständig. Es fehlen die Flexionen einiger Namen. Namen werden vor Orten erkannt, denn die Ortsliste ist übervollständig und enthält u.a. viele Namen. Die Ortsliste wurde aus dem vollständigen sorbischen Lexikon erstellt unter Entfernung aller kleingeschriebenen Wörter, Abkürzungen und EinwohnerInnen von Orten (erkennbar an den Endungen '-čan' und '-čanka'). Würden die Namen nicht vor den Orten erkannt, dann würden alle Namen als Orte erkannt.
Weiterhin wird der Name des 16-Einwohner-Dorfes Haj aus der Gemeinde Radwor deutlich öfter erkannt als gemeint, da 'haj' zugleich 'ja' bedeutet.

3.3.3 Teilwortzerlegung

3.3.3.1 BPE

Byte Pair Encoding (BPE) ist ein Komprimierungsalgorithmus, der häufig bei der Verarbeitung natürlicher Sprache und zur Datenkompression eingesetzt wird. BPE ersetzt häufig auftretende Texte mit kürzeren Codes, wodurch eine effiziente Speicherung und Verarbeitung von Text möglich ist.

BPE verbindet iterativ die häufigsten Zeichenfolgen des Eingabetextes, ersetzt sie mit einem Code und legt die Code-Text-Paare in einem Wörterbuch ab. Diese Schritte werden wiederholt, bis eine maximale Anzahl an Codes oder eine gewünschte Kompression erreicht wurde.

Dieses Verfahren wurde in vielen NLP-Aufgaben eingesetzt, bspw. für maschinelle Übersetzung, Textklassifikation und Eigennamenerkennung,

3.3.3.2 Morfessor

Morfessor umfasst eine Reihe von unüberwachten Lernalgorithmen zur morphologischen Zerlegung von Wörtern für die Verarbeitung natürlicher Sprache. Die Grundidee ist eine datengetriebene Wortzerlegung in kleinere, bedeutungstragende Einheiten (engl. Morphs), ohne vordefinierte Regeln oder Vorwissen über die Sprache zu benötigen.

Der Morfessor-Algorithmus startet mit einer initialen Menge von Morphs. Iterativ werden sie verbunden oder aufgeteilt basierend auf statistischen Messgrößen bezüglich ihres Auftretens in den Eingabedaten. Das Ergebnis ist eine Menge von Morphs, welche die häufigsten und am stärksten bedeutungstragenden Teilwörter der Sprache enthalten.

Morfessor wurde bei zahlreichen NLP-Aufgaben angewandt, wie zum Beispiel maschinelle Übersetzung, Spracherkennung und Text-zu-Sprache-Synthese.

Segmentierung der Wörter in morphologische Einheiten durch einen automatischen Parser und Überprüfung der Aufteilung anhand der Aussprache der Wörter und Einheiten:

> Für die Segmentierung wurde die Python-Bibliothek "Morfessor" der Version 2.0 benutzt. Hierfür wurde ein Python Skript geschrieben zum automatischen Einlesen von Textdateien, die anschließend zu Training der Modelle von Morfessor genutzt wurden. Mithilfe dieser Modelle wurde anschließend die Eingabetexte segmentiert und eine Liste der vom Modell erkannten Wortsegmente erstellt. Die trainierten Modelle, sowie die Segmentierung werden anschließend gesondert abgespeichert.

Erzeugung eines Vokabulars und eines Lexikons für die morphologischen Einheiten:

> Für die Erzeugung des Vokabulars und der morphologischen Einheiten, wurde ein Programm geschrieben, welches die segmentierten Texte verarbeitet und die einzelnen Segmente nach ihrer möglichen Position im Wort gesondert auflistet.

3.3.4 Evaluation

3.3.4.1 Teilwortzerlegung

Die Leistung des Sprachmodells wurde mittels drei Text Korpora aus den verschiedenen Domänen evaluiert. Die folgende Textressourcen kamen für das Training (unüberwacht und überwacht) von den BPE und Morfessor Modellen zum Einsatz.

Als Metriken zur Bewertung wurden verwendet: der relative Unterschied bei der Vergrößerung des Korpus (engl. corpus size increase, weniger ist besser), die Reduktion der Anzahl der Token (engl. unique token decrease, mehr ist besser) und die durchschnittlichen Tokenlänge (engl. average token length decrease, weniger ist besser).

Beim Training der Modelle für die Tokenization wurden der `hsb` Korpus und `golden_standard.vocab` verwendet:

- `hsb:` umfassendster Textkorpus, bestehend aus:
  - HSB Common Voice v5.1
  - sorbian_institute_monolingual
  - web_monolingual
  - witaj_monolingual
  - adaptation
  - V4.1 (filmy_Normiert, lektorizowane_hs_teksty_bobr_Normiert, myto_cisinskeho_Normiert, wselcizny_Normiert)
- `golden:` kombiniertes Vokabular von HSB, Lexeme und Soblex

Korpus	Vokabulargröße	Mittlere Tokenlänge
HSB	389.408	9,40
Golden	3.025.155	12,22

Abb. 3 Ressourcen für das Training, Umfang des Vokabulars und ihre durchschnittliche Tokenlänge..

	Anzahl Sätze	Anzahl Token	Anzahl versch. Tokens	Mittlere Tokenlänge
BIBLE_NER	33.626	759.037	53.565	8,03
MCN_NER	892	14.566	5.251	7,83
SPW_NER	5.114	78.472	10.073	7,02
HSB	591.101	6.025.845	389.408	9,40
KALDI_V2	29.162	167.996	9.471	7,54

Abb. 4 Korpora zur Evaluation und ihre Eigenschaften.

Korpus	Teilwortzerlegung	Corpus size increase (%)	Unique tokens decrease (%)	Average token length decrease (%)
BIBLE_NER	morfessor_golden	-1	-3,19	-1,76
	morfessor_hsb	**39,23**	-57,99	**-22,9**
	bpe_golden_5000_5	88,79	**-83,88**	-42,7
	bpe_hsb_5000_5	70,09	-83,15	-42,47
	bpe_golden_10000_5	75,8	-75,69	-39,89
	bpe_hsb_10000_5	57,6	-74,6	-39,52
HSB	morfessor_gold	4,26	-7	-1,75
	morfessor_hsb	47,23	-70,17	**-22,72**
	bpe_golden_5000_5	121,24	**-96,39**	-51,69
	bpe_hsb_5000_5	97,03	-96,26	-51,2
	bpe_golden_10000_5	104,2	-93,59	-48,42
	bpe_hsb_10000_5	80,99	-93,39	-47,94
KALDI_V2	morfessor_gold	2,22	1,86	-1,88
	morfessor_hsb	**39,42**	-20,93	**-17,08**
	bpe_golden_5000_5	108,41	**-47,84**	-39,61
	bpe_hsb_5000_5	86,13	-43,62	-38,64
	bpe_golden_10000_5	93,92	-33,66	-37,02
	bpe_hsb_10000_5	70,34	-27,48	-35,73
MCN_NER	morfessor_gold	2,2	3,56	-2,91
	morfessor_hsb	**44,43**	-13,83	**-19,04**
	bpe_golden_5000_5	116,03	**-88,00**	-42,2
	bpe_hsb_5000_5	89,78	-21,12	-40,95
	bpe_golden_10000_5	98,71	-14,57	-39,68
	bpe_hsb_10000_5	75,39	-7,87	-37,8
SPW_NER	morfessor_gold	-3,03	0,57	-1,88
	morfessor_hsb	**28,78**	-24,5	**-17,95**
	bpe_golden_5000_5	77,83	**-53,2**	-36,87
	bpe_hsb_5000_5	58,95	-49,87	-35,94
	bpe_golden_10000_5	64,81	-41,36	-34,63
	bpe_hsb_10000_5	45,51	-36,05	-33,31

Abb. 5 Ergebnisse der Evaluation je Korpus und Tokenization-Modell.

In den Ergebnissen zeigte sich, dass das Vokabular von `golden` in Kombination mit Morfessor ungeeignet ist.
Das liegt wahrscheinlich an dem Format (einfache Liste von Wörtern) und an der großen Anzahl Wörter mit ihren Lexemen, weshalb der Algorithmus zugrundeliegende Muster nicht verlässlich bestimmen konnte.
Stattdessen wäre ein großer Textkorpus mit verschiedenen Wortformen und Zusammenhängen besser geeignet (bspw. `hsb.corp`).
Jedoch bi `golden_standard.vocab` gute Trainingsdaten für den BPE Algorithmus. Für die nächsten Schritte werden die Modelle `bpe_golden_5000_5` und `morfessor_hsb` verwendet.

3.3.4.2 Sprachmodelle

Das Training der Sprachmodelle erfolgte mit einer zufällig gewählten Stichprobe, die 95% aller Sätze umfasste. Die Reproduzierbarkeit der Ergebnisse wurde durch das Festsetzen des Seeds sichergestellt. Die restlichen 5% der Sätze wurden für die Evaluation der Modelle verwendet.
Für kleine Korpora führte das zu häufigeren Verwechslungen, weil die Testdaten unbekannte N-Gramme oder Wörter außerhalb des Vokabulars enthielten.

Bei den Experimenten zur Spracherkennung wurden alle Sätze der Korpora berücksichtigt und die Sprachmodelle damit trainiert.

Abb. 6 Perplexität und Anzahl Wörter außerhalb des Vokabulars (OOV) bezüglich der Sprachmodelle.

Korpus	Teilwortzerlegung	Test OOV	3-gram		4-gram		5-gram	
			Train	*Test*	*Train*	*Test*	*Train*	*Test*
BIBLE_NER	none	1.347	8,53174	477,816	3,56166	482,351	3,43529	482,106
	morfessor_hsb	331	11,5772	122,083	4,38588	121,923	3,40414	122,811
	bpe golden 5000	**56**	**12,0463**	**42,2794**	**5,21398**	**41,0759**	**3,65068**	**41,3524**
	bpe golden 10000	133	11,8421	53,7995	4,95589	52,8723	3,57894	53,3714
	bpe hsb 5000	56	11,6716	60,84	4,63699	59,743	3,48121	60,0295
	bpe hsb 10000	135	11,2263	81,9281	4,40542	81,2542	3,42664	81,661
HSB	None	11.474	24,7327	884,714	17,3873	882,438	17,2918	882,397
	morfessor_hsb	1.896	28,0604	195,501	18,1571	191,081	17,799	190,384
	bpe golden 5000	**58**	**23,4144**	**43,4152**	**17,8091**	**39,7053**	**16,9609**	**39,0606**
	bpe golden 10000	136	24,381	56,9734	18,1283	53,0447	17,3727	52,4554
	bpe hsb 5000	53	26,2093	62,5402	18,9342	58,5109	18,3679	57,9205
	bpe hsb 10000	108	26,2532	86,8973	18,7055	82,894	18,2937	82,3639
KALDI_V2	None	227	4,70422	6,77701	4,1901	6,05709	4,16974	6,03403
	morfessor_hsb	128	3,97013	6,30301	3,1326	5,0457	3,04566	4,93004
	bpe golden 5000	**39**	**3,63203**	**5,56848**	**2,53246**	**4,01134**	**2,30716**	**3,69293**
	bpe golden 10000	84	3,60146	5,52263	2,58062	4,07493	2,39883	3,80578
	bpe hsb 5000	45	3,35604	**5,50276**	2,56818	4,28717	2,45206	4,11789
	bpe hsb 10000	79	3,40213	5,6057	2,68375	4,51746	2,60229	4,39396

Korpus	Teilwortzerlegung	Test OOV	3-gram		4-gram		5-gram	
			Train	*Test*	*Train*	*Test*	*Train*	*Test*
MCN_NER	None	184	3,43003	223,273	2,92383	221,972	2,88389	222,024
	morfessor_hsb	125	3,55644	175,442	2,69242	175,528	2,5931	175,198
	bpe golden 5000	**60**	**3,94564**	**86,183**	**2,72195**	**84,1461**	**2,42887**	**84,0407**
	bpe golden 10000	86	3,78281	105,02	2,67301	103,855	2,44121	103,516
	bpe hsb 5000	68	3,53366	138,581	2,61602	137,503	2,45322	136,87
	bpe hsb 10000	106	3,47886	161,048	2,61987	159,688	2,48735	159,667
SPW_NER	None	398	4,36684	208,63	3,07037	204,846	2,97759	205,23
	morfessor_hsb	249	5,32306	104,235	3,06311	102,957	2,70995	103,1
	bpe golden 5000	**82**	**6,02418**	**46,0242**	**3,20962**	**44,8053**	**2,57824**	**44,8984**
	bpe golden 10000	117	5,88388	57,9797	3,1255	56,4714	2,58737	56,642
	bpe hsb 5000	95	5,45609	69,2609	3,03509	67,1991	2,58671	67,3134
	bpe hsb 10000	145	5,25201	88,1801	2,96854	85,9345	2,61662	86,1442

Es ist offensichtlich, dass kleinere Tokenvokabulare zu einer geringeren Perplexität führen. Jedoch erhöhen kurze Tokens die akustische Verwechselbarkeit. Deshalb empfiehlt sich für das Sprachmodell die Betrachtung eines größeren Kontexts und die geringere Filterung der Wörter (engl. Pruning). Dies würde jedoch auch zu einer schlechteren WER führen, wenn die Domänen nicht übereinstimmen, weil sich der modellierte Kontext nicht in der Zieldomäne wiederfindet.

3.4 Zusammenfassung

Die Verarbeitungsschritte und notwendige Funktionen zur Vorverarbeitung elektronischer Textdaten wurden erstellt. Dadurch war es möglich, eine Sammlung von normalisierten Textkorpora zu erzeugen, der für das Training von statistischen Sprachmodellen basierend auf ganzen Wörtern oder Wortteilen geeignet ist. Eine Eigennamenerkennung zur Wortklassenmodellierung mit FST-Grammatiken wurde ebenso implementiert.

4 AP 3: Spracherkennung

4.1 Übersicht

Das Open-Source-Erkennungsmodul aus den frühehren Projekten (`dLabPro/UASR`) ist besser geeignet für Anwendungen mit klaren Steuerungsbefehlen (engl. Command-and-Control, kurz C&C), endlichen Grammatiken (FSG) und einem begrenzten Vokabular. In diesem Fall kann der Erkenner nicht für LVCSR-Anwendungen eingesetzt werden.
Der maßgeschneiderte Fhg IKTS Erkenner, steht nicht als Open-Source-Lösung zur Verfügung.

4.2 Zielstellungen

Es soll eine vorkompilierte Erkennungsanwendung (Bibliotheken mit Beispielquellcode) für Intel 64 Bit und Linux OS bereitgestellt werden. Ihr Einsatz soll ausschließlich auf den IT-Geräten und Servern der „Sorbischen Stiftung“ und keinen Dritten zur Verfügung stehen.

Der Erkenner beschränkt sich auf Sprachanwendungen mit den Sprachen Ober- und Niedersorbisch.

Der Erkenner akzeptiert Audiodaten oder vorberechnete Merkmale und liefert die Erkennungsergebnisse.

Es sollen sich verschieden Modelle (Akustische Modelle, Lexikon, Sprachmodelle) einsetzen und konfigurieren lassen.

Die Spracherkennungsmethoden aus den früheren Projekten sollen für die Merkmalberechnungen von Audiodateien modifiziert werden.

4.3 Ergebnisse

Es entstand eine Anwendung für die Spracherkennung mit Hilfsfunktionen und Skripte (Erkenner, Konfigurator) mit einem Benutzerhandbuch.
Modifikationen der Open-Source-Bibliotheken (`dLabPro/UASR`) für die Offline-Feature-Extraktion von Audiodateien wurden vorgenommen.

4.4 Übersicht

Eine proprietäre Spracherkennung und Konfigurationsmöglichkeiten wurden entwickelt und zur Integration in Sprachanwendungen bereitgestellt. Die Erkennungssoftware unterstützt und ist teilweise kompatibel mit häufig genutzten Open-Source-Tools für akustische Modelle, Lexika und Sprachmodelle.

5 LVCSR-Anwendungsentwicklung

5.1 Evaluation

Das (GPM) Modell wurde im Folgenden beispielhaft untersucht mit `kaldi_v2.corp` und einer zugehörigen Domäne. Insbesondere stimmten die N-Gramme in den Trainings- und Testdaten überein und es existierten keine Wörter außerhalb des Vokabulars.

Modell	Teilwortzerlegung	COR	ERR	LD
3-gram	none	89,8	14,4	102,8
4-gram	none	90,6	13,5	102,9
3-gram	bpe_gold_5000_5	78,0	26,6	101,3
4-gram	bpe_gold_5000_5	83,6	20,4	101,6
5-gram	bpe_gold_5000_5	84,9	18,9	101,7

Abb. 7 Korrektheit (COR in %), Fehlerrate (ERR in %) und Label-Dichte (LD in %); Ergebnisse für CV Datensatz mit kaldi_v2.corp Korpus.

Die Erkennungsqualität wurde mit der Wortfehlerrate (WER) evaluiert, welche sich mit der Levenshtein-Distanz bzgl. der Referenztransliterationen errechnen lässt.

Die Unterschiede dieser Ergebnisse im Vergleich zu den KALDI-Ergebnissen lassen sich größtenteils auf die verschieden Decodierungsparametern zurückführen. Die LD zeigt, dass das Worteinfügegewicht (WIP) angepasst werden sollte und sodass weniger Wörter bei der Decodierung erkannt werden, um eine bessere Echtzeiterkennung zu erreichen.

Es ist anzumerken, dass die Teilwort-basierten Modelle schlechtere Ergebnisse liefern. Dies liegt an den kürzeren Tokens und den kleineren Kontext der Sprachmodelle.

Für die (DSM) Modelle wurden die Korpora `Bible + MCN + SPW` zu einem `gd_ner.corp` zusammengefasst und damit ein Sprachmodell trainiert und die Perplexität bei den `misa230319` Transliterationen ausgewertet.

Anschließend wurde die Erkennungsleistung auf den Audioausschnitten von den `misa230319` YouTube-Videos ermittelt.

Wir kombinierten verschiedene Modelle, um unbekannten Kontext und OOVs zusammenzufügen. Die Perplexitätsberechnung wurde vor und nach der Sprachmodellkombination durchgeführt.

Im Korpus `misa.corp waren 434 OOVs`. Dies deutet auf eine fehlende Übereinstimmung der Domänen hin und führt zu einer deutlich schlechteren Spracherkennung. Im Vergleich waren im Teilwortvokabular nur fünf OOVs.

Testkorpus	Sprachmodell (ARPA)	Perplexität	OOV
gs_ner.corp	gd_ner.corp_all_3	8,60892	0
	merged	4,23049	0
misa.corp	gd_ner.corp_all_3	878,925	434
	merged	0,802868	0
gd_ner_bpe_golden_5000_5.corp	gd_ner_bpe_golden_5000_5.corp_all_3	12,2941	0
	merged_bpe	5,98013	0
misa_bpe_golden_5000_5.corp	gd_ner_bpe_golden_5000_5.corp_all_3	77,4882	5
	merged_bpe	0,904831	0
gd_ner_mrph_hsb.corp	gd_ner_mrph_hsb.corp_all_3	11,7211	0
	merged_morph	3,86616	0
misa_morph.corp	gd_ner_mrph_hsb.corp_all_3	498,167	170
	merged_morph	1,10091	0

Abb. 8 Perplexität und OOVs für das vollständige, BPE und Morfessor Token-basierte Modell.

Es ist zu sehen, dass das kombinierte Modell eine bessere Perplexität bei beiden Quellkorpora besitzt (bei den Teilwörtern).

Modell	Teilwortzerlegung	COR	ERR	LD
3-gram	none	45,6	59,4	97,6
4-gram	none	45,7	58,9	96,9
3-gram	morfessor_hsb	43,6	61,5	98,2
3-gram	bpe_gold_5000_5	39,1	66,4	97,7
4-gram	bpe_gold_5000_5	42,2	64,0	98,9
5-gram	bpe_gold_5000_5	43,2	62,9	99,4

Abb. 9 Korrektheit (COR in %), Fehlerrate (ERR in %) und Label-Dichte (LD in %); Ergebnisse für misa230319 Datensatz mit gd_ner.corp Korpus.

Modell	Teilwortzerlegung	COR	ERR	LD
3-gram	none	89,0	18,7	106,3
3-gram	morfessor_hsb	83,0	23,6	104,8
3-gram	bpe_gold_5000_5	54,3	59,9	111,5

Abb. 10 Korrektheit (COR in %), Fehlerrate (ERR in %) und Label-Dichte (LD in %); Ergebnisse für misa230319 Datensatz mit 3-Gramm kombinierten Modellen.

Obwohl sich die Fehlerrate bei den Teilwortmodellen verbesserte, spiegelte das nicht die signifikante Verbesserung der Perplexität wider. Die verhältnismäßig kurzen Tokens verursachten eine zu starke akustische Verwechselbarkeit. Außerdem reichen 3-Gramme nicht für die erfolgreiche Modellierung des Kontexts innerhalb der Wörter.

Die Label-Dichte macht deutlich, dass es noch Raum für Verbesserungen gibt, indem Bestrafung für das inkludieren weiterer Wörter und die Gewichte des Sprachmodells optimiert werden.

Die Pruning-Parameter spielen bei der Decodierung eine sehr wichtige Rolle. Hohe Werte führen bei den Teilwort-basierten Modelle zu einer signifikant längeren Laufzeit der Erkennung.

Das Ausbalancieren zwischen Anzahl Wörter (Tokens) im Vokabular und ihrer durchschnittlichen Länge ist sehr wichtig. Das Training von Morfessor- oder BPE-Modellen mit mehr Tokens und größeren N-Grammen in der Zieldomäne ist in der Praxis empfehlenswert.

5.2 Zusammenfassung

Die Ergebnisse lassen sich in den folgenden Punkten zusammenfassen:

- Teilwort-basierte Modellierung reduziert die Anzahl von OOVs und verbessert die Perplexität des Sprachmodells.
- Teilwort-basierte Modellierung ist ein Kompromiss zwischen dem Umfang des Vokabulars (möglichst klein) und der Länge von Tokens (möglichst lang).
- Das Interpolieren von Sprachmodellen verbessert die Leistung für beide Domänen der Quellkorpora.
- Hyperparamter müssen optimiert auf der Zieldomäne mit passenden Sprachmodellen optimiert werden.
- Eine akzeptable Performanz kann nur erreicht werden, wenn der Textkorpus mit der Zielsprachdomäne übereinstimmt, wie es bspw. bei dem KALDI-Korpus und den interpolierten Modellen (ohne Tokens und mit Morfessor Tokens) zu sehen war.

6 Schlussfolgerung und zukünftige Arbeiten

Die erstellten Sprach- und Textressourcen sind wertvolle Assets, die zur Digitalisierung und Erhaltung der obersorbischen Sprache beitragen.

Die gesammelten Erfahrungen, entwickelten Prozeduren und die entstandene Software bilden die Grundlage für weitere Anwendungen in den Bereichen der Computerlinguistik und Sprachtechnologie (Sprachsynthese, Dialogmanagement, Verstehen und Generieren natürlicher Sprache und Maschinenübersetzung). Außerdem ermögliche sie die schnellere Einbindung neuer Daten, die zur weiteren Verbesserung der Modelle genutzt werden können.

Zukünftige Arbeiten sollen sich auf die Verbesserung und Optimierung der Technologien konzentrieren, indem neue Daten gesammelt werden.

Außerdem sollen weitere Möglichkeiten für moderne Technologien und neuartigen Sprachanwendungen untersucht werden. Bereiche für deren Einsatz sind bspw. Bildung, Büro und Verwaltung, Live-Untertitel und -Übersetzung von Medien, persönlicher virtueller Assistent und Chatbots, Verbesserung der Lebensqualität für Personen mit Behinderungen.