Zürcher Hochschule
für Angewandte Wissenschaften

School of
Engineering
CAI Centre for
Artificial Intelligence

ZÜRCHER HOCHSCHULE FÜR ANGEWANDTE
WISSENSCHAFTEN

Improved Speech Translation for Swiss German using a hybrid Dynamic Window Approach

---

Verfasser:
Kevin Kläger
Lukas Bamert

Betreuer:
Prof. Dr. Mark Cieliebak
Dr. Jan Milan Deriu

Bachelorarbeit

10. Juni 2022

Erklärung betreffend das selbstständige Verfassen einer Bachelorarbeit an der School of Engineering

Mit der Abgabe dieser Bachelorarbeit versichert der/die Studierende, dass er/sie die Arbeit selbständig und ohne fremde Hilfe verfasst hat. (Bei Gruppenarbeiten gelten die Leistungen der übrigen Gruppenmitglieder nicht als fremde Hilfe.)
Der/die unterzeichnende Studierende erklärt, dass alle zitierten Quellen (auch Internetseiten) im Text oder Anhang korrekt nachgewiesen sind, d.h. dass die Bachelorarbeit keine Plagiate enthält, also keine Teile, die teilweise oder vollständig aus einem fremden Text oder einer fremden Arbeit unter Vorgabe der eigenen Urheberschaft bzw. ohne Quellenangabe übernommen worden sind.

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

Ort, Datum:	**Name Studierende:**
Mosnang, 10.06.2022	Kevin Kläger
Winterthur, 10.06.2022	Lukas Bamert

Zusammenfassung

Die Forschung arbeitet seit mehreren Jahren intensiv an Speech Translation Systemen, die schweizerdeutsche Sprache zu deutschem Text übersetzen sollen. Das Fehlen einer offiziellen Grammatik sowie die Vielzahl an Dialekten stellen dabei grosse Herausforderungen an ein Speech Translation System. Die meisten Systeme für Schweizerdeutsch werden aktuell auf satzweise Übersetzung trainiert und ausgewertet. In der Realität sind Audioaufnahmen aber meistens mehrere Minuten lang und beinhalten mehr als einen Satz / Sprecher. Mit einem System, das darauf trainiert wurde einen Satz zu übersetzen, kann man deswegen nicht direkt längere Sequenzen ohne Qualitätseinbussen übersetzen. Mit genau diesem Problem beschäftigt sich diese Arbeit. Mittels synthetischer Datensätze werden längere Szenarien simuliert. Um die Audiodatei aufzuteilen, verfolgt die erhaltene Pipeline einen Sliding Window Ansatz. Wir zeigen, dass durch intelligentes Preprocessing mit Algorithmen wie Speech Activity Detection oder Speaker Diarization die Übersetzungsqualität auf längeren Sequenzen um bis zu 125% auf 58.19 BLEU Punkte gesteigert werden kann. Dies ist schon sehr nahe an der theoretischen Referenz, bei der jeder Satz einzeln übersetzt wird und damit 59.69 BLEU Punkte erreicht werden. Bei noch schwierigeren Sequenzen ohne Sprechpause, bei denen die Audiodatei nicht mehr einfach aufgeteilt werden kann, belegen wir, dass ein dynamischer Window Ansatz die Übersetzungsqualität verdoppelt. Auf dem komplexeren Podclub Datensatz, der aus Podcastaufnahmen besteht, verbessert sich der BLEU Score mit einer Kombination aus Preprocessing und dynamischem Window Ansatz, dem hybriden Dynamic Window Ansatz, um 64% auf 41.57 Punkte. Eine qualitative Analyse zeigt aber, dass das Übersetzungssystem noch nicht genügend gute Übersetzungen liefert, die für automatische Transkriptionen ausreichen.

Abstract

Researchers have been working intensively for several years on Speech Translation Systems, which are intended to translate Swiss German into German text. The lack of an official grammar, as well as the multitude of dialects, pose great challenges for a Speech Translation System. Most systems for Swiss German are currently trained and evaluated for sentence-by-sentence translation. However, audio recordings are usually several minutes long and contain more than one sentence / speaker. With a system trained to translate one sentence, it is therefore not possible to translate longer sequences directly without a loss of quality. This work deals with exactly this problem. Synthetic data sets are used to simulate longer scenarios. To split the audio file, the resulting pipeline follows a sliding window approach. We show that by intelligent preprocessing with algorithms such as Speech Activity Detection or Speaker Diarization, the translation quality on longer sequences can be increased by up to 125%, to 58.19 BLEU points. This is already very close to the theoretical reference, where each sentence is translated individually and thus 59.69 BLEU points are achieved. For even more difficult sequences without pauses in speech, where the audio file can no longer be simply split, we prove that a dynamic window approach doubles the translation quality. On the more complex Podclub dataset, which consists of podcast recordings, the BLEU score improves by 64% to 41.57 points with a combination of preprocessing and dynamic window approach, the hybrid dynamic window approach. However, a qualitative analysis shows that the translation system does not yet provide good enough translations for automatic transcriptions.

Vorwort

Eigeninteresse, sowie die Bereitschaft ein herausforderndes und relevantes Thema bearbeiten zu wollen sind die Gründe, wieso wir diese Arbeit gewählt haben. Für die Unterstützung sowie die wertvollen Inputs während der gesamten Arbeit möchten wir unseren Betreuern Prof. Dr. Mark Cieliebak und Dr. Jan Milan Deriu danken. Fragen wurden schnell beantwortet und die Diskussionen waren immer sehr hilfreich und zielgerichtet.
Weiter möchten wir uns bei Karin Iselin, Sakana Iselin und Urs Bamert für das Korrekturlesen bedanken. Ausserdem richten wir einen speziellen Dank an Felicia Aepli, die uns mit ihrer Expertise bei der Erstellung der Visualisierungen unterstützt hat.

Inhaltsverzeichnis

Kapitel 1

Einleitung

Protokollieren von Meetings, das Erstellen von Untertiteln für Filme und Generieren von Übersetzungen sind nur einige Aufgaben, die heutzutage mittels Speech Translation (ST) Systemen automatisiert werden. Im Vergleich zum ähnlichen Forschungsgebiet der Automatic Speech Recognition (ASR) unterscheidet sich dabei die Quellsprache von der Zielsprache. In beiden Bereichen wurden in den letzten Jahren enorme Fortschritte erzielt, welche sich durch eine steigende Anzahl an Datensätzen und Rechenleistung sowie neuen Erkenntnissen aus der Forschung erklären. Mittlerweile gibt es eine Vielzahl an Anbietern, darunter Microsoft [1], Amazon [2] und Google [3], die für viele Sprachen, darunter auch Schweizerdeutsch, bereits Lösungen anbieten. Bei Sprachen, von denen eine grosse Menge an Trainingsdaten existieren, sind die Übersetzungen bereits sehr gut. Bei ressourcenärmeren Sprachen wie Schweizerdeutsch sind die Übersetzungsqualitäten jedoch bedeutend schlechter. Neben dem, dass Schweizerdeutsch zu den ressourcenärmeren Sprachen zählt, wird die Übersetzung zusätzlich durch die Vielzahl an Dialekten und der Inexistenz einer formal verankerten, geschriebenen Sprache erschwert. Nichtsdestotrotz wurden in den letzten Jahren grosse Fortschritte im Bereich ST von Schweizerdeutsch auf Hochdeutsch gemacht. Im Jahr 2021 wurde von SwissText ein Wettkampf, Shared Task 3, durchgeführt, an dem unter anderem Microsoft und die ZHAW teilgenommen haben [4]. Arabskyy et al. [5], die Gewinner des Wettbewerbs, haben dabei auf ein hybrides System gesetzt, das auf einem Graph-To-Phoneme Model (G2P) und einem akustischem Modell, einer Art LSTM, basiert [5]. Auf dem Datensatz des Shared Task haben sie damit einen BLEU Score von 46 Punkten erreicht. Ein Jahr später stellten Plüss et al. [6] zusammen mit einem neuen Korpus bereits ein Modell vor, das auf dem SDS-200 Datensatz einen BLEU Score von 64 erreicht [6]. Dieses Modell wiederum basiert auf einem vortrainierten XLS-R [7] Modell. BLEU Scores von verschiedenen Korpora sind zwar nicht direkt vergleichbar, es lässt sich aber innerhalb von nur einem Jahr trotzdem eine starke Verbesserung feststellen.

1.1 Problemstellung

Damit ST oder ASR Systeme für reale Anwendungen eingesetzt werden können, müssen sie Audiodateien verabeiten können, die mehr als einen Satz beinhalten. Das Problem ist aber, dass sie meistens auf einzelnen Sätzen trainiert wurden und mehrere Sätze damit nicht trennen können. Desweiteren können gängige Modelle maximal zwanzig Sekunden in einem Schritt verarbeiten. Bei ASR Anwendungen, bei der die Quellsprache die gleiche ist wie die Zielsprache, kann meistens Wort für Wort übersetzt werden. Dadurch genügt ein einfacher Sliding Window Ansatz mit Überlappung, um die Übersetzung zu generieren [8]. Bei ST Systemen spielt der lokale Kontext jedoch eine weit wichtigere Rolle, da teilweise pro Satz der Satzbau und die Grammatik verändert werden muss, um die Satzlogik zu erhalten. Die Übersetzung von Schweizerdeutscher Sprache zu Hochdeutsch bildet dabei mit ihren zahlreichen Dialekten keine Ausnahme.

1.2 Zielsetzung

Die Arbeit baut auf einer ST Pipeline auf, die wie das vorgestellte Modell von Plüss et al. [6] auf einem vortrainierten XLS-R Modell [7] basiert. Dieses wurde auf der Schweizer Dialektsammlung [6] finegetuned und besitzt eine Milliarde Parameter. Für die Verarbeitung von längeren Audiodateien benützt sie einen Sliding Window Ansatz. Das Ziel der Arbeit ist, die Übersetzungsqualität der bestehenden Pipeline auf längeren Audiodaten mit mehreren Sätzen zu verbessern. Das ST Modell kann dabei nicht weitertrainiert werden. Es werden verschiedene Ansätze getestet, um Audiodaten bestmöglich aufzuteilen. Des Weiteren wird ein erweitertes Sliding Window sowie ein dynamischer Window Ansatz evaluiert, der auf Sentence Scoring basiert.
Abschliessend werden Ansätze kombiniert.

1.3 Related Work

Für ASR Systeme gibt es bereits Ansätze, womit Satzgrenzen gut erkannt werden können. Biron et al. [9] achten dabei auf Sprechpausen sowie Unstetigkeiten in der Sprechgeschwindigkeit, genauer das Verlängern von Silben vor den Satzgrenzen und die Beschleunigung an Satzanfängen. Eine weitere Arbeit achtet auf andere akustische Eigenschaften und kombiniert diese mit einem "boundary confidence coefficient", der

mittels Language Model auf dem ASR Ouput berechnet wird [10]. Bei beiden Ansätzen wird jedoch bereits der Output vom ASR System benötigt. Tsiamas et al. [8] und Gaido et al. [11] beschreiben in ihren Arbeiten den grossen Einfluss des Aufteilens von Audiodaten an den Satzgrenzen vor der Übersetzung durch ein ST System, womit die oben genannten Ansätze nicht einfach kopiert werden können. Tsiamas et al. [8] präsentieren einen Ansatz, bei dem ein vortrainiertes wav2vec2.0 [12] Modell auf Boundary Detection von Sätzen trainiert wird und dann mit Hilfe eines Divide&Conquer Algorithmus Satzgrenzen definiert werden [8]. Da das Paper während des Bearbeitens dieser Arbeit veröffentlicht wurde, konnte der Ansatz aus zeitlichen Gründen aber nicht auf Schweizerdeutsch trainiert und getestet werden. Weitere Arbeiten, wie die von Gaido et al. [11], versuchen die Audiodatei vor der Übersetzung mittels eines hybriden Segmentierungsalgorithmus aufzuteilen. Dieser teilt das Audio innerhalb von Intervallen einer minimalen und maximalen Länge bei der längsten Sprechpause auf. Falls im Intervall keine Sprechpause gefunden wird, wird das Segment bei der maximalen Länge des Intervalles aufgeteilt. Die Sprechpausen werden mittels Voice Activity Detection (VAD), auch Speech Activity Detection (SAD) genannt, ermittelt. Auch in dieser Arbeit werden Ansätze verfolgt, die das Audio zuerst mittels SAD oder Speaker Diarization aufteilen. Falls hierbei keine Segmentierung gefunden wird, werden andere Ansätze verfolgt. Das Intervall wird dann mittels Sliding Window oder einem Dynamic Window Ansatz aufgeteilt. Der Dynamic Window Ansatz ist ein Algorithmus, der das Intervall in verschiedene Segmente aufteilt und anschliessend die generierten Übersetzungen für diese Segmente evaluiert und den besten Kandidaten auswählt.

1.4 Methodik

Um reale Anwendungszwecke zu simulieren, wurden verschiedene synthetische Datensätze erzeugt. Damit wurden drei Szenarios entworfen. Das erste Szenario beinhaltet längere Audiosequenzen mit Sprechpausen, das zweite Audiosequenzen mit mehreren Sätzen vom selben Sprecher ohne Sprechpausen. Das letzte Szenario soll Dialoge zwischen jeweils zwei Personen ohne Sprechpause simulieren. Als Basis für die synthetischen Datensätze diente ein Datensatz, in dem pro Audiodatei genau ein Satz gesprochen wird. Dies hatte den Vorteil, dass als Referenz die einzelnen Sätze isoliert übersetzt werden konnten. Somit war es möglich, einen Referenzscore, beschrieben in Kapitel 6.3.2, zu berechnen, welcher ungefähr der maximalen Übersetzungsqualität der ST Pipeline entspricht. Danach wurden verschiedene Experimente mit dem Ziel gemacht, den BLEU-Score auf diesen Szenarien zu maximieren. Durch die drei Szenarien, die jeweils leicht unterschiedliche Herausforderungen an ein ST System stellen,

konnte anschliessend mit Hilfe des Referenzscores nachvollzogen werden, welche Herausforderungen die verschiedenen Experimente gut oder schlecht meistern. Durch die Erkenntnisse aus diesen Analysen wurden anschliessend Experimente verbessert und kombiniert.

1.5 Aufbau der Arbeit

In der nachfolgenden Arbeit werden zuerst Grundlagen zu verschiedenen Themen erläutert, die in dieser Arbeit von Bedeutung sind. Die Auswahl der Tools wird im nachfolgenden Kapitel beschrieben. Zusätzlich wird argumentiert, weshalb man sich für diese Tools entschieden hat. Danach werden verschiedene Datensätze vorgestellt, auf denen in der Arbeit Experimente und Evaluationen durchgeführt wurden. Im darauf folgenden Kapitel wird erklärt, wie synthetische Datensätze aus bestehenden erstellt werden, um mehrere definierte Szenarien zu simulieren. Als nächstes wird die ST Pipeline (Baseline Pipeline), welche für diese Arbeit von Dr. Jan Milan Deriu zur Verfügung gestellt wurde, erläutert, evaluiert und analysiert. Im Anschluss werden Experimente zur Verbesserung vorgestellt sowie ausgewertet. Eine komplette Übersicht aller Resultate findet sich im Anschluss an die Experimente. Das letzte Kapitel schliesst die Arbeit mit einer kritischen Diskussion der Resultate und einem Ausblick auf mögliche zukünftige Experimente ab. Die Arbeit richtet sich an Fachexperten, die bereits über Wissen in den Grundlagen von Machine Learning besitzen.

Kapitel 2

Grundlagen

In diesem Kapitel werden theoretische Grundlagen erläutert, welche für die Arbeit essentiell sind. Es wird ein Überblick über die Thematik gegeben, damit nachfolgende Kapitel verständlicher sind. Zusätzlich wird der aktuelle Stand in der Forschung, der "state-of-the-art", beschrieben.

2.1 Speech Translation

Mithilfe von Speech Translation (ST), auch Automatic Speech Translation (AST) genannt, wird das Audio von einer Sprache zu Text einer anderen Sprache übersetzt [13]. Im Gegensatz zu Speech To Text (STT), auch Automatic Speech Recognition (ASR) genannt, muss das Modell zusätzlich einen Übersetzungsschritt trainieren, wodurch sich die Aufgabe bedeutend komplexer gestaltet [13].

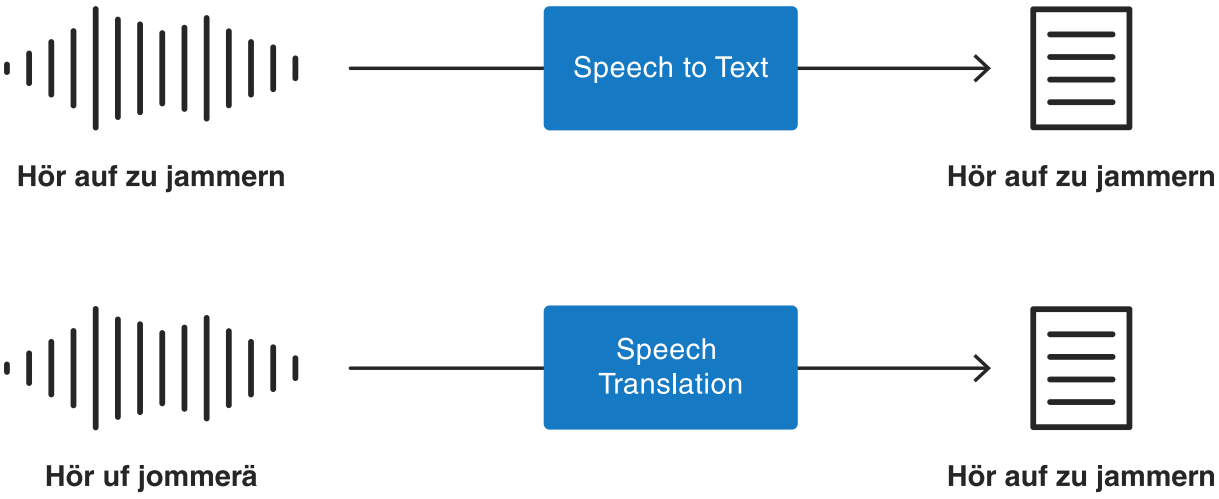


ABBILDUNG 2.1: Darstellung der Aufgaben von Speech To Text (oben) und Speech Translation (unten).

Arbeiten wie Untertitel in Filmen/Videos, Transkription von Meetings oder dolmetschen, können mit ST automatisiert werden. Je nach Aufgabe ändern sich aber die Anforderungen an ein ST System. Sperber und Paulik [13] unterscheiden beispielsweise folgende Anwendungen:

- **Batch Mode**. Beim Batch Mode wird jeweils ein ganzes Stück Audio in einem Schritt eingelesen und verarbeitet. Nach der Verarbeitung wird die Übersetzung ausgegeben. Die Antwortzeiten spielen hierbei keine Rolle. Als Beispiele hierfür werden Filmuntertitel oder Film Dubbing genannt.
- **Consecutive**. Beim Consecutive Modus werden übersetzbare Stücke mit möglichst kurzer Antwortzeit übersetzt. Ein Beispiel dafür wäre eine Übersetzungsapp für das Smartphone.
- **Simultaneous**. Im Simultaneous Modus wird ein Audiostream verarbeitet und in Echtzeit übersetzt. In diesem Setting ist eine schnelle Antwortzeit essentiell.

Diese Arbeit handelt von einem ST System, das im Batch Mode übersetzt.

Die momentan besten state-of-the-art ST Systeme basieren auf Transformern, genauer wav2vec2 Modellen [12]. Dabei wurde von Babu et al. [7] im Jahr 2021 ein mehrsprachig vortrainiertes Wav2Vec Modell präsentiert, das nach Finetuning gleich gute Ergebnisse liefert wie ein zuvor monolingual trainiertes. Dies ist sehr interessant für Schweizerdeutsche Übersetzungssysteme, da das Modell anschliessend auf ressourcenarmen Sprachen finegetuned werden kann.

2.2 Language Model

Ein Language Model (LM) ist ein Modell, das Wahrscheinlichkeiten für Sequenzen von Wörtern vergibt [14]. Es existieren verschiedene Arten von Language Modellen, wobei in dieser Arbeit genauer auf n-gram und neuronale Language Modelle eingegangen wird.

2.2.1 GPT-2

Generative Pre-trained Transformer 2 [15], kurz GPT-2, ist ein transformerbasiertes Language Model. Das Modell wurde für die Aufgabe trainiert, aufgrund von vorangehendem Text das nächste Wort vorherzusagen. Dabei erzeugt es aber Wahrscheinlichkeiten für jedes Wort im Vokabular, weshalb es für Natural Language Processing Aufgaben breit eingesetzt werden kann [15]. Mit GPT-3 [16] existiert bereits ein Nachfolger von GPT-2, dessen Source Code jedoch nicht veröffentlicht wurde. Es steht jedoch eine kostenpflichtige API zur Verfügung, mit der man das Modell nutzen könnte [17].

Für diese Arbeit wurde ein Open Source GPT-2 Modell verwendet, das auf 16 GB deutschem Text trainiert wurde und eine Vokabulargrösse von 50k besitzt [18].

2.2.2 N-Gram

Ein n-gram ist eine Sequenz von Wörtern der Länge n. Ein Beispiel für ein 2-gram wäre "Guten Tag". Um die Wahrscheinlichkeit, dass ein gewisses Wort nach einer Sequenz vorkommt, zu berechnen, werden jeweils die letzten $n-1$ Wörter in Betracht gezogen. Aus den vom Training erhaltenen Daten wird dann die relative Wahrscheinlichkeit berechnet, dass dieses Wort nach dieser Sequenz auftritt [14]. Für diese Arbeit wurde uns ein 5-gram Model zur Verfügung gestellt, das von Dr. Jan Milan Deriu auf deutschem Text von Wikipedia trainiert wurde. Im Unterschied zum neuronalen Netz werden nur die letzten vier Wörter in Betracht gezogen. Ausserdem werden keine Abstraktionen gelernt, sondern nur die Anzahl Auftretungen gezählt und für die Berechnung verwendet.

2.3 Evaluationsmetriken für Speech Translation

Es existieren unterschiedliche Bewertungsmetriken, um die Genauigkeit einer Übersetzung zu bestimmen [19]. Beispiele für Metriken sind BLEU [20], WER, METEOR [21] und viele mehr. Dabei gibt es keine Metrik, die grundlegend besser als alle anderen ist und immer genutzt wird [19]. Nachfolgend werden die Evaluationsmetriken genauer beschrieben, die in dieser Arbeit eingesetzt werden.

2.3.1 Word Error Rate

Word Error Rate (WER) misst die Genauigkeit der Übersetzung Wort für Wort. Die Metrik berechnet sich aus den Substitutionen (S), Einfügungen (E), Löschungen (L), der Anzahl korrekter Wörter (C) und der Anzahl Wörter in der "Ground-Truth" (N). Die WER liegt zwischen 0 und 1, wobei 0 bedeutet, dass die Übersetzung der Ground-Truth entspricht.

$$WER = \frac{S + L + E}{N}$$

2.3.2 BLEU Score

Grundsätzlich existieren meistens mehrere "perfekte" Übersetzungen eines Textes. Bilingual Evaluation Understudy (BLEU) ist ein Algorithmus, der die Ähnlichkeit einer maschinellen Übersetzung zu einer oder mehreren hochwertigen Referenzübersetzungen misst [22]. Das Ziel dieser Metrik ist es, die Übersetzung menschlicher zu bewerten, als dieses zum Beispiel die Word Error Rate macht [20]. Grundsätzlich wird das damit erreicht, dass n-grams einer Übersetzung mit den n-grams einer oder mehrerer Referenzen verglichen werden. Mit der Anzahl an Übereinstimmungen werden für verschiedene n jeweils n-gram Präzisionen berechnet. Zusätzlich wird für jeden Satz ein "Sentence Brevity Penalty" berechnet, der zusätzlich die Länge, Wortwahl und Wortreihenfolge der Übersetzung in Betracht zieht. Der Brevity Penality multipliziert mit dem geometrischen Mittel aller n-gram Präzisionen resultiert im BLEU Score [20].

Da die Anzahl Referenzübersetzungen und die Wahl des Korpus grosse Einflüsse auf den BLEU Score haben kann, wird davon abgeraten, diesen Korpus übergreifend und mit abweichender Anzahl Referenzübersetzungen zu vergleichen. Grundsätzlich kann der BLEU Score folgendermassen interpretiert werden.

BLEU-Wert	Interpretation
< 10	Fast unbrauchbar
10-19	Schwierig, das Wesentliche zu verstehen
20–29	Das Wesentliche ist verständlich, aber es gibt erhebliche Grammatikfehler
30-40	Verständliche bis gute Übersetzungen
40-50	Hochwertige Übersetzungen
50-60	Sehr hochwertige, adäquate und flüssige Übersetzungen
> 60	Qualität oft besser als menschliche Übersetzungen

TABELLE 2.1: Interpretation Bleu Score gemäss [22]

2.4 Speech Activity Detection

Das Ziel von Speech Activity Detection (SAD) ist, Audiosegmente mit Sprechaktivität von Audiosegmenten ohne Sprechaktivität zu unterscheiden [23]. SAD wird auch Voice Activity Detection (VAD) genannt und ist unter anderem bei komplexeren Aufgaben wie Speaker Diarization, beschrieben in Kapitel 2.5, im Einsatz.

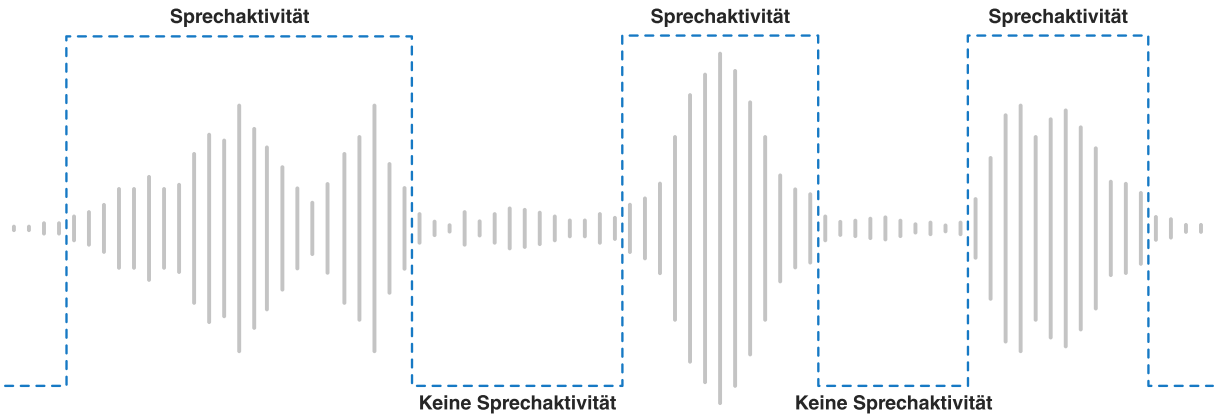


ABBILDUNG 2.2: Visualisierung der Aufgabe von Speech Activity Detection (SAD).

Für SAD existieren zahlreiche Modelle mit verschiedenen Lösungsansätzen. Dabei verzeichnen Deep Neural Network (DNN) Ansätze state-of-the-art Ergebnisse [24, 25]. Ein aktuelles state-of-the-art Modell [25] benutzt ein Convolutional Neural Network

(CNN) [26] mit Bidirectional Long Short-Term Memory (BiLSTM) [27] und verzeichnet eine Area Under the ROC Curve (AUC) von 0.951. Wilkinson und Niesler [25] vergleichen ihre SAD mit anderen Systemen, wobei der zweite Rang eine AUC von 0.8564 erreicht.

2.4.1 Area Under the ROC Curve

Die Area Under the ROC Curve (AUC) [28] misst die Fläche der sogenannten Receiver Operating Characteristic (ROC) Kurve. Die ROC Kurve zeigt dabei die Performance des Modells für alle Thresholds auf. Sie existiert nur für binäre Klassifikationsaufgaben. Für jeden Threshold wird die True Positive Rate (TPR), sowie die False Positive Rate (FPR) gemessen. Die Resultate davon werden dann in einer Grafik dargestellt. Die nachfolgende Grafik zeigt mehrere potentielle ROC Kurven und illustriert, wie eine ROC Kurve zu interpretieren ist.

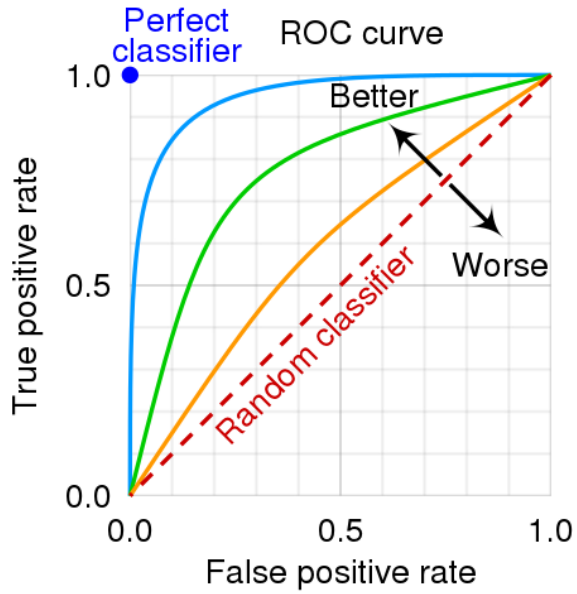


ABBILDUNG 2.3: Interpretation ROC Kurve inklusive Darstellung verschiedener ROC Kurven. Die AUC ist jeweils die Fläche unter den Kurven. Das Bild ist von [29]

2.5 Speaker Diarization

Das Ziel von Speaker Diarization ist das Herausfinden von Sprecherintervallen. Also wer wann gesprochen hat [23]. Dabei kann die Anzahl der Personen bekannt oder unbekannt sein. Wird Speaker Diarization mit Automatic Speech Recognition kombiniert, kann daraus ermittelt werden, welche Person was gesagt hat. Ein Anwendungsfall dafür wäre die automatische Transkription von Meetings.

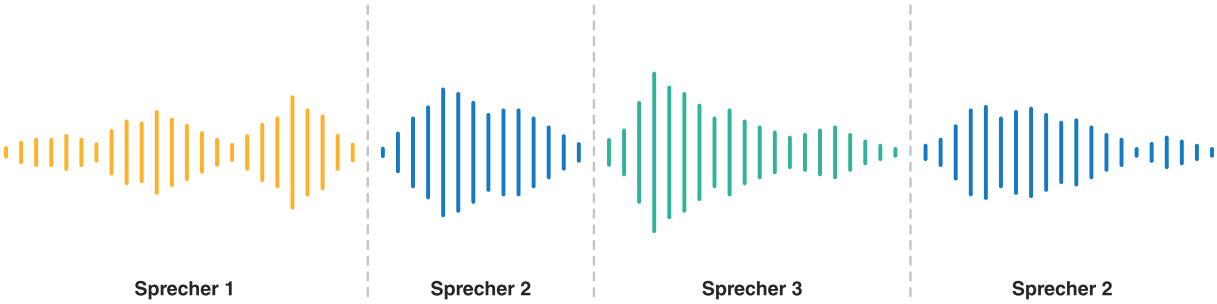


ABBILDUNG 2.4: Visualisierung der Aufgabe eines Speaker Diarization Systems.

Es existieren unterschiedliche Architekturen und Herangehensweisen für Speaker Diarization [23]. Wie in Abbildung 2.5 visualisiert werden Speaker Diarization Systeme oftmals als eine sequentielle Anordnung einzelner Module implementiert. Die Abgrenzung der Module ist dabei nicht fest definiert und variiert je nach Anwendungsfall.

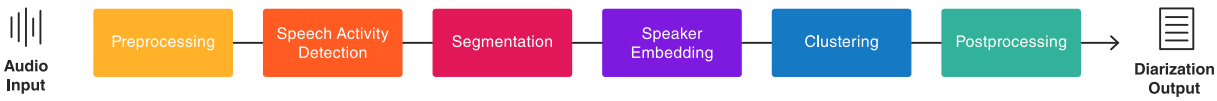


ABBILDUNG 2.5: Mögliche Architektur eines Speaker Diarization Systems aufgeteilt in einzelne Module.

Aktuelle Modelle mit state-of-the-art Leistung basieren auf DNN [23, 30, 31]. Modelle wie UIS-RNN [30] erreichen auf dem CALLHOME Datensatz [32] eine Diarization Error Rate (DER) von 7.6 Prozent. Das sind 1.2 Prozentpunkte mehr wie bisherige Modelle, die auf Spectral Clustering basieren (8.8 Prozent auf dem gleichen Datensatz).

2.5.1 Diarization Error Rate

Diarization Error Rate (DER) ist die Standard-Metrik, um die Performance von Speaker Diarization Systemen zu messen [23]. DER ist das Verhältnis von falsch zugewiesenen

Sprachsegmenten zu der totalen Anzahl von Sprachsegmenten, unter der Annahme, dass alle Sprachsegmente gleich lang sind. DER ist folgendermassen definiert [23]:

$$DER = \frac{FA + M + C}{AnzahlSegmente}$$

FA: False Alarm - Segment als Sprecher definiert, obwohl 'Kein Sprecher' vorhanden war
M: Missed - Segment als 'Kein Sprecher' definiert, obwohl ein Sprecher vorhanden war
C: Speaker-Confusion - Falscher Sprecher definiert

2.6 Connectionist Temporal Classification

Connectionist Temporal Classification (CTC) wird benutzt um Modelle zu trainieren, die als Eingang Sequenzen bekommen und deren Ausgang wiederum eine Sequenz ist. Dazu gehören auch Speech-To-Text oder Übersetzungsmodelle. Das Labelling wird weiter dadurch erschwert, dass die beiden Sequenzen nicht gleich lang sein müssen. Ein weiteres Problem beim Lernen von Sequenzen ist, dass Buchstaben mehrmals gehört werden können. Ein Modell kann beispielsweise "TOR" als "TOOORR" erkennen. Als Output liefert das Modell eine Matrix bestehend aus den Wahrscheinlichkeiten jedes Buchstabens zu jedem Zeitpunkt der Ausgangssequenz. Die Aneinanderreihung von Buchstaben für jeden Zeitpunkt wird als Pfad bezeichnet. Der Wert eines Pfades berechnet sich aus der Multiplikation der Wahrscheinlichkeiten für jeden Buchstaben b_x zum Zeitpunkt t_x.
Die resultierenden Pfade werden zudem decodiert, das heisst, dass doppelte Buchstaben entfernt werden. Falls ein Buchstabe doppelt hintereinander vorkommen soll, muss das also mit einem speziellen Buchstaben gekennzeichnet werden.

Um den Loss zu berechnen werden nun die Werte aller Pfade summiert, die decodiert dem Wert der Ground-Truth entsprechen [33].

2.7 Beam Search

Der Beam-Search Algorithmus funktioniert ähnlich wie die Greedy-Search, besitzt aber zusätzliche Variablen und Kontext. Die Anzahl Kandidaten wird beschränkt durch den

Parameter Beam Size. Der Algorithmus startet beim Zeitpunkt t_0 und wählt die Buchstaben mit den höchsten Wahrscheinlichkeiten aus und fügt diese jedem Kandidaten als ersten Buchstaben zu. Bei den weiteren Schritten werden dann jeweils für jeden Kandidaten die bereits gewählten Buchstaben in Betracht gezogen und es werden wiederum die wahrscheinlichsten Kandidaten ausgewählt [34].

2.8 wav2vec2

Wav2vec2 ist ein Framework um self-supervised Kontextdarstellungen von rohen Audiodaten zu trainieren, welche dann mittels CTC auf gelabelten Daten für ASR oder ST Aufgaben finegetuned werden können. Das Modell basiert auf einem mehrlagigen feature encoder, der für jeden Zeitpunkt aus dem rohen Audio mittels Convolution Sprachrepräsentationen erzeugt. Diese werden wiederum in ein Transformernetzwerk weitergeleitet, das für jeden Zeitpunkt Kontextdarstellungen erzeugt. [12]

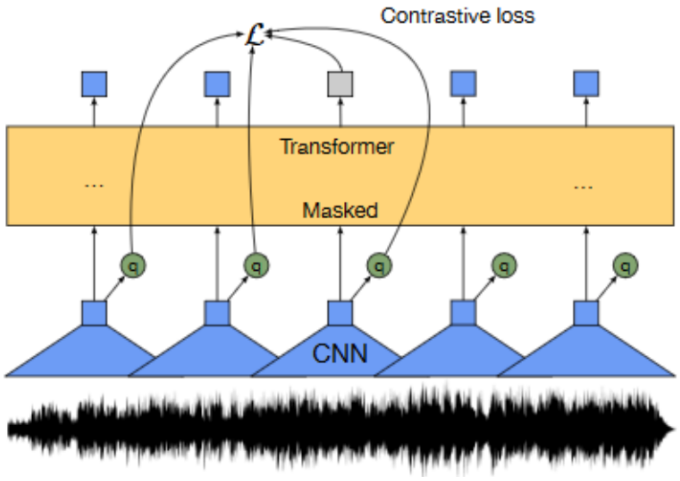


ABBILDUNG 2.6: Wav2Wec2.0 Modell [12]

Das Self-Training wird mittels Maskierung eines Teils des Outputs des Feature Encoders erreicht. Die maskierten Bereiche werden versucht vorherzusagen. Mit den Vorhersagen kann dann wiederum ein Loss berechnet werden. Vortrainierte Modelle für ASR und ST Aufgaben haben zusätzlich einen linearen Layer über dem Transformernetzwerk, der die Kontextdarstellungen in die Anzahl Tokens im Vokabular projiziert. Laut Alexei Baevski übertrifft dieser neue Ansatz alle vorhergängigen state-of-the-art Methoden im Bereich von Self-Training, sogar wenn hundertmal weniger ungelabelte mit gleich vielen gelabelten Daten zur Verfügung stehen. [12]

Kapitel 3

Verwendete Tools

In diesem Kapitel wird die Auswahl der Tools beschrieben, welche für diverse Aufgaben innerhalb der Arbeit verwendet wurden. Weiter soll dargelegt werden, weshalb man sich für diese entschieden hat.

3.1 Programmiersprache

Python wurde als Programmiersprache für diese Arbeit ausgewählt. Zum einen ist sie die dominante Sprache in der Domäne künstlicher Intelligenz. Zum anderen basiert der zur Verfügung gestellte Code bereits auf Python.

3.2 Transformerframework

Als Transformerframework wurde Huggingface [35] verwendet. Huggingface bietet nicht nur eine komplette Opensource Library für verschiedenste Aufgaben bezüglich künstlicher Intelligenz, sondern es werden auch direkt Datensätze und vortrainierte Modelle frei zugänglich gemacht [36]. Die erhaltene Pipeline baut bereits auf Huggingface auf, weshalb ein anderes Tool nicht zur Diskussion stand.

3.3 Audio Verarbeitung

Um Audios laden und verarbeiten zu können, wurde librosa [37] eingesetzt. Librosa wurde ausgewählt, da es eine grosse Community besitzt (über 5000 Github Sterne)

sowie viele nützliche Funktionen zur Audioverarbeitung bietet. Zusätzlich ist es in der Bedienung sehr einfach und besitzt eine gute Dokumentation. Des Weiteren benutzt der zur Verfügung gestellte Code bereits librosa für die Audioverarbeitung.

3.4 Evaluationstools

3.4.1 BLEU score

Um den BLEU-Score zu berechnen, wurde "sacrebleu" [38] benutzt. Die Library hat über 500 Github Sterne und wird häufig eingesetzt. Zudem war die Einbindung sehr einfach. Es werden die Standardeinstellungen von sacrebleu genutzt. Dabei wird auf Gross-/Kleinschreibung geachtet.

3.4.2 Word Error Rate

Für die WER Berechnung fiel die Wahl auf JiWER [39], weil es für WER Berechnungen in Python beliebt ist und einfach eingebunden werden konnte.

3.5 Speech Activity Detection

Als Tool für SAD wurde pyannote.audio [40] verwendet. Das Toolkit besitzt über 1500 Github Sterne und konnte einfach eingebunden werden. Pyannote.audio liefert zusätzlich mehrere Modelle für verschiedene Aufgaben (SAD, Speaker Diarization etc.). Die Ergebnisse einer ersten qualitativen Analyse waren zudem sehr zufriedenstellend.

3.6 Speaker Diarization

Für Speaker Diarization wurde das Modell von pyannote verwendet, da pyannote bereits für SAD benutzt wurde. Zudem schreibt Bredin et al. [31], dass die Speaker Diarization von pyannote.audio eine gute Leistung (kleiner DER Score) erzielt.

Durch einen Tipp von Dr. Jan Milan Deriu wird zusätzlich zu pyannote.audio eine interne Diarization API der ZHAW verwendet, welche mittels REST API angesprochen werden kann.

3.7 Evaluationsumgebung

Alle Auswertungen wurden auf mehreren Instanzen vom Openstack Cluster der ZHAW [41] durchgeführt. Eine Cluster Instanz hat dabei 16GB Arbeitsspeicher sowie acht virtuelle CPUs vom Typ Intel Core Processor (Broadwell, no TSX, IBRS) mit 3Ghz Leistung und eine nVidia Tesla T4 GPU Karte [42]. Auf jeder Instanz läuft als Betriebssystem Ubuntu 20.04.4 LTS mit nVidia CUDA Version 11.6. Die Ressourcen mussten während dieser Arbeit nicht mit einer anderen Person geteilt werden.

Kapitel 4

Datensätze

In diesem Kapitel werden verschiedene schweizerdeutsche Datensätze vorgestellt, die für diese Arbeit verwendet wurden. Das Ziel dieses Kapitels ist es, eine Übersicht der Eigenschaften pro Datensatz zu geben.

4.1 Schweizer Dialektsammlung

Die "Schweizer Dialektsammlung" [43] ist ein Projekt, in der beliebige Nutzer über eine Webapplikation Audioaufnahmen in Schweizerdeutsch aufnehmen und überprüfen können. Das Ziel des Projektes ist es, einen Datensatz zu erschaffen, mit dem Modelle für Aufgaben wie ST trainiert werden können. Der Datensatz wurde im Verlauf dieser Arbeit veröffentlicht [6].

4.1.1 Aufbau Datensatz

Der Datensatz enthält über 140’000 Audioaufnahmen von knapp 4’000 verschiedenen Sprechern, die jeweils einen Satz in Schweizerdeutsch sprechen. Die Aufnahmen besitzen dabei eine Länge zwischen 2 und 12 Sekunden, wie in der folgenden Abbildung zu sehen ist. Im Durchschnitt ist eine Audiodatei 4.8 Sekunden lang. Insgesamt beinhaltet der Datensatz über 200 Stunden Audiodaten inklusive der Transkription der Sätze auf Hochdeutsch. Die Audioqualität der verschiedenen Aufnahmen variiert stark.

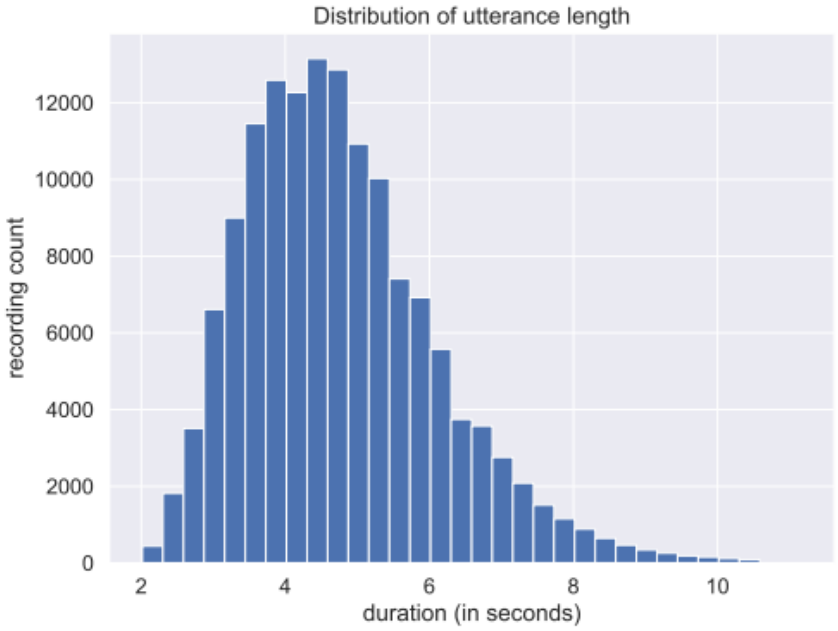


ABBILDUNG 4.1: Histogramm Audiolängen der Schweizer Dialektsammlung. Ausschnitt aus dem Paper von Plüss et al. [6].

4.2 SNF - Schweizerischer Nationalfonds

Beim SNF Datensatz handelt es sich um einen internen Datensatz der ZHAW, der bisher nicht veröffentlicht wurde. Er ist ähnlich aufgebaut wie die Dialektsammlung.

4.2.1 Aufbau Datensatz

Der Datensatz enthält über 25'000 Audioaufnahmen von 76 verschiedenen Sprechern, die jeweils einen Satz in Schweizerdeutsch sprechen. Wie in der folgenden Abbildung erkennbar haben die Aufnahmen eine Länge von 2 bis 15 Sekunden. Im Durchschnitt ist eine Audiodatei 4.96 Sekunden lang. Insgesamt beinhaltet der Datensatz 34 Stunden Audiodaten inklusive der Transkription der Sätze auf Hochdeutsch. Die Aufnahmen sind von guter Qualität und sehr gut verständlich.

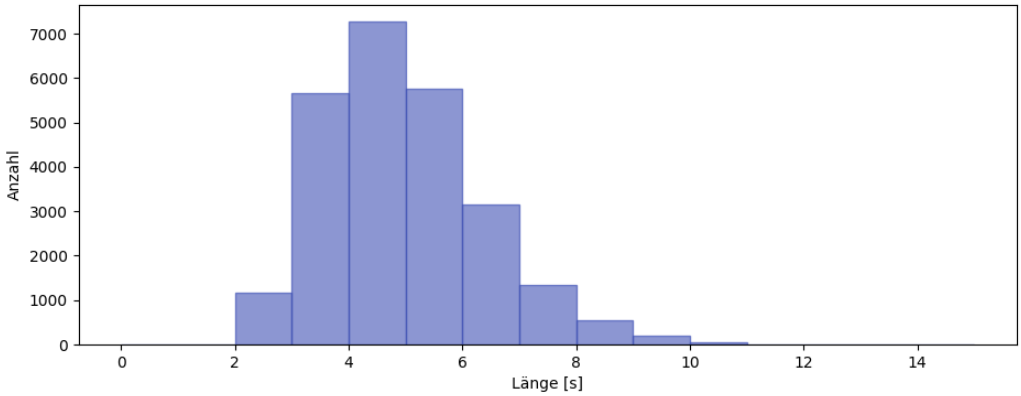


ABBILDUNG 4.2: Histogramm Audiolängen für das Testset des SNF Datensatzes.

4.3 SwissText 2021 Testset - All Swiss German Dialects Test Set

Im Wettbewerb des SwissText 2021 Task 3 [4] ging es darum, ein ST Modell zu erstellen, das einzelne Sätze von Schweizerdeutsch (Audio) zu Hochdeutsch (Text) übersetzen kann. Es gibt einen "public" und einen "private" Teil des Datensatzes, wobei der private Teil für das finale Ranking entscheidend war [4]. Im weiteren Verlauf der Arbeit wird der private Teil des Datensatzes "SwissText 2021 Testset" kurz "SwissText" genannt.

4.3.1 Aufbau Datensatz

Der Datensatz enthält 2'785 Audioaufnahmen von 178 verschiedenen Sprechern, die jeweils einen Satz in Schweizerdeutsch sprechen. Die Aufnahmen besitzen dabei eine Länge zwischen 2 und 20 Sekunden, was in der folgenden Abbildung visualisiert wird. Im Durchschnitt ist eine Audiodatei 7.94 Sekunden lang. Insgesamt beinhaltet der Datensatz 6 Stunden Audiodaten, inklusive der Transkription der Sätze auf Hochdeutsch. Die Audioaufnahmen besitzen eine gute Qualität und sind frei von Hintergrundgeräuschen.

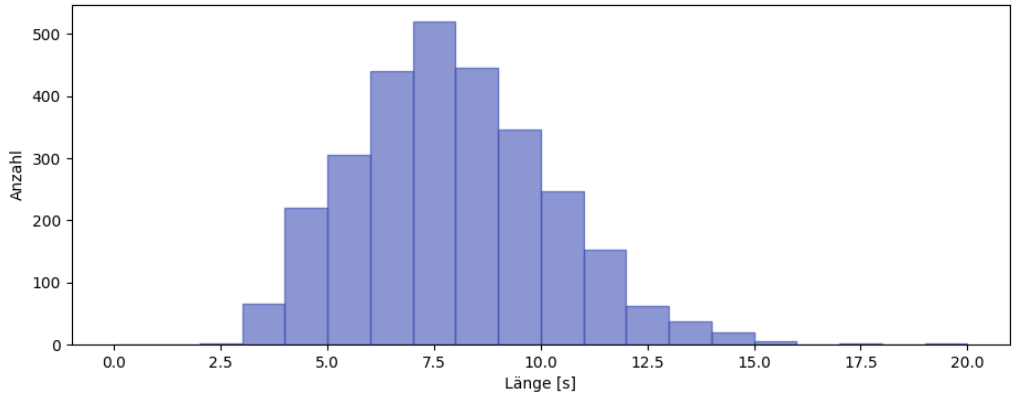


ABBILDUNG 4.3: Histogramm Audiolängen für SwissText 2021 Testset.

4.4 Podclub

Der Podclub Datensatz ist eine nicht veröffentlichte Sammlung von Sprachpodcasts, welche die ZHAW gesammelt hat. Mithilfe der Aufnahmen können Personen verschiedene Sprachen üben. Dieser Podclub Datensatz enthält nur Aufnahmen in Schweizerdeutsch.

4.4.1 Aufbau Datensatz

Die Audioaufnahmen sind zwischen 8 und 12 Minuten lang. Es spricht jeweils eine Person und erzählt darin eine Geschichte. Insgesamt enthält der Datensatz über 13 Stunden Audioaufnahmen. Jede Audioaufnahme besitzt eine Transkription auf Hochdeutsch. Die Audioaufnahmen besitzen eine gute Qualität und sind frei von Hintergrundgeräuschen.

Damit der Datensatz zur Evaluation genutzt werden konnte, musste er noch bearbeitet werden.
Zum einen wurden die Intros / Outros im Audio weggeschnitten, weil die Transkripte diese nicht beinhalten. Das Intro ist bei allen Aufnahmen 28 Sekunden lang und konnte einfach weggeschnitten werden. Es gibt aber zwei unterschiedliche Outros (28 Sekunden und 50 Sekunden lang). Hier musste manuell festgelegt werden, wie lang das Outro pro Audio ist und dementsprechend geschnitten werden.
Zum anderen mussten die Transkripte vereinfacht werden, da sie Wortdefinitionen am

Ende, Absätze, Zeilenumbrüche und Sonderzeichen beinhalteten. Diese wurden aus den Transkripten gelöscht.

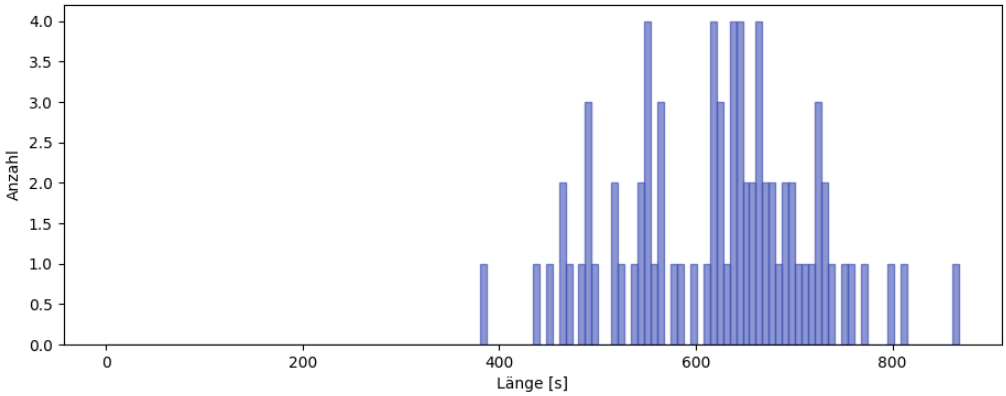


ABBILDUNG 4.4: Histogramm Audiolängen für den Podclub Datensatz.

Kapitel 5

Synthetische Daten

Um Anwendungsfälle zu simulieren, die mit den bestehenden Datensätzen nicht nachgestellt werden konnten, wurden diese Daten synthetisch erzeugt. Als Basis für die synthetischen Datensätze dient der SNF Datensatz (siehe Kapitel 4.2). Dieser bot sich an, da er aus einzelnen Sprachaufnahmen mit jeweils einem Satz besteht und nicht zum Training des Modells verwendet wurde. Die synthetischen Datensätze werden in Szenarien unterteilt, die jeweils einen Anwendungsfall abbilden sollen. Dieses Kapitel beschreibt die Erzeugung und Eigenschaften der verschiedenen Szenario-Datensätze.

5.1 Szenario 1: Lange Audiodatei

Bei diesem Szenario handelt es sich um Audioaufnahmen, die aus mehr als nur einem Satz/Sprecher bestehen und mindestens zwei Minuten dauern. Zwischen den verschiedenen Sätzen werden jeweils kurze Pausen gemacht.

5.1.1 Aufbau Datensatz

Bei der Erzeugung einer Audioaufnahme wurden automatisch einzelne Sätze hintereinander kopiert, bis die Aufnahme mindestens 120 Sekunden lang war. Die Audioaufnahme wurde anschliessend als WAV Datei gespeichert. Wichtig zu erwähnen ist, dass die verschiedenen Sätze zusammenhangslos sind und als Gespräch keinen Sinn ergeben.

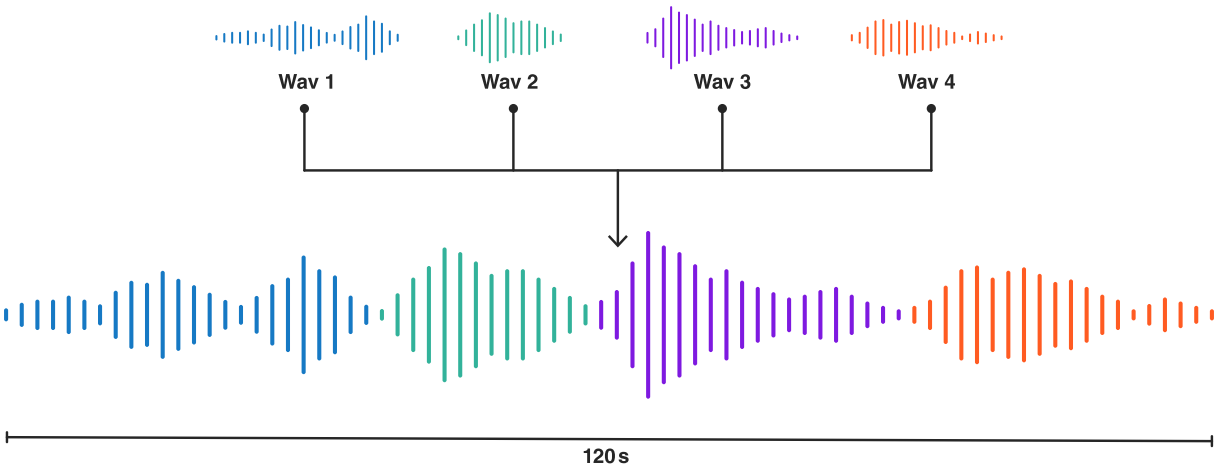


ABBILDUNG 5.1: Visualisierung der Erstellung einer Audiodatei des Szenario 1 Datensatzes.

Insgesamt wurden 200 solcher Dateien generiert mit einer Gesamtlänge von etwa sieben Stunden.

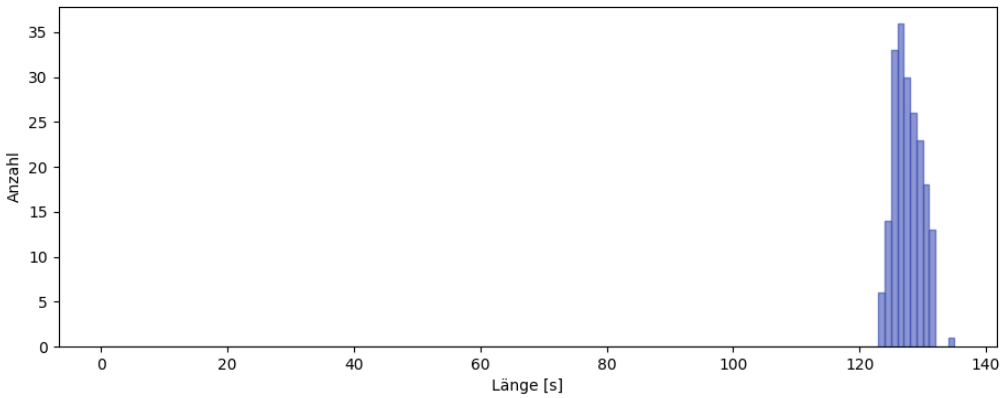


ABBILDUNG 5.2: Histogramm Audiolängen für Szenario 1: Lange Audiodatei.

5.1.2 Metadaten

Zusätzlich zum Audio wurden folgende Metadaten pro Aufnahme gespeichert:

- *index*: Referenz zur Audiodatei.
- *text*: Transkript zur Audiodatei. Setzt sich aus den Texten der einzelnen Audioaufnahmen zusammen.
- *time_slots*: Start und Ende (in Sekunden) der einzelnen Audio Aufnahmen im neuen Audio, welche kopiert wurden.
- *clip_ids*: Originale Ids der einzelnen Audioaufnahmen aus dem SNF Datensatz.

5.2 Szenario 2: Keine Sprechpausen

In diesem Szenario handelt es sich um Audioaufnahmen, bei denen ein einzelner Sprecher ohne Pause über eine längere Zeit spricht. Die Dauer einer Aufnahme beträgt mindestens 30 Sekunden.

5.2.1 Aufbau Datensatz

Für die Erstellung wurden von den Aufnahmen jeweils nur die Intervalle berücksichtigt, welche Sprechanteile enthalten. Es wurden also die Silence-Segmente am Anfang und Ende jedes Satzes entfernt. Anschliessend wurden diese geschnittenen Sätze für einen Sprecher hintereinander kopiert, bis die Gesamtlänge der resultierenden Audiosequenz mindestens 30 Sekunden betrug. Die Audiosequenz wurde daraufhin als WAV Datei gespeichert. Für die Ermittlung der Sprechintervalle wurde das SAD Modell von pyannote.audio [40] verwendet. Es wurden nur Audioaufnahmen für diesen Datensatz berücksichtigt, welche im SAD Output genau eine Sprechzeit beinhalten. Somit werden Audioaufnahmen ausgeschlossen, in denen eine Sprechpause während des Satzes gemacht wird. Damit von den Aufnahmen möglichst der ganze Satz zu hören ist, wurde die Sprechzeit, die vom SAD Modell definiert wurde, zusätzlich um 0.05 Sekunden am Start und Ende erweitert. Diese Massnahme soll verhindern, dass das erste und letzte Wort teilweise abgeschnitten werden.

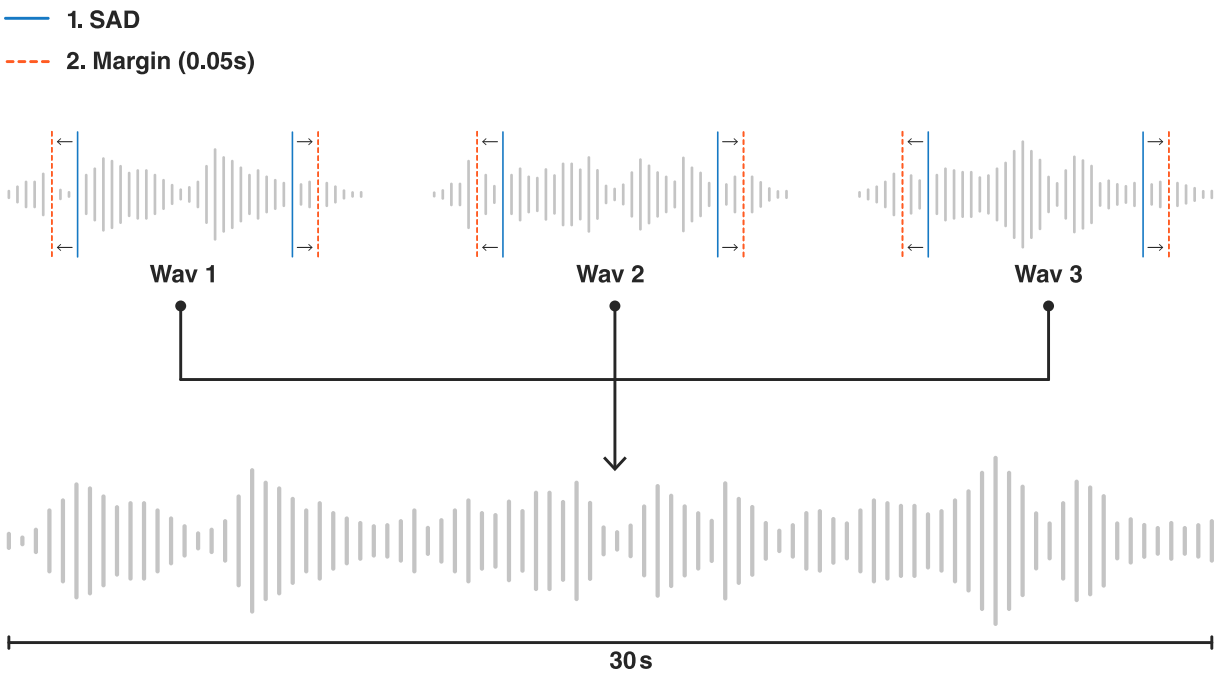


ABBILDUNG 5.3: Visualisierung der Erstellung einer Audiodatei des Szenario 2 Datensatzes.

Es wurde für jeden Sprecher im SNF Datensatz, der genügend Audiodaten aufgenommen hat, eine Audiodatei erstellt. Insgesamt wurden also für 75 Sprecher eine Audiodatei an mindestens 30 Sekunden erstellt. Dies resultiert in einer Gesamtlänge von etwas mehr als 30 Minuten. Zusätzlich zu den Audiodateien wurden ausserdem die Transkriptionen festgehalten.

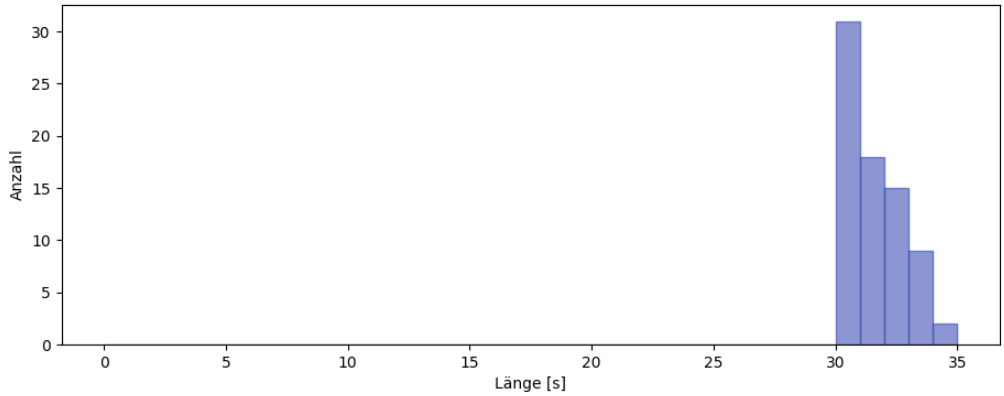


ABBILDUNG 5.4: Histogramm Audiolängen für Szenario 2: Lange Sprechzeiten.

5.2.2 Metadaten

Die Metadaten sind identisch zu Kapitel 5.1.2.

5.3 Szenario 3: Dialog

Das Szenario Dialog besteht aus Audioaufnahmen, bei denen ein Sprecher einen Satz spricht und direkt im Anschluss ein zweiter Sprecher eine "Antwort" gibt. Die Aufnahmen sollen einen Dialog simulieren, wobei es zwischen den beiden Sätzen keine Pause gibt.

5.3.1 Aufbau Datensatz

Die Erzeugung der Audioaufnahmen funktioniert ähnlich wie in Szenario 2 in Kapitel 5.2. Auch hier wurden für jeden Satz nur die Intervalle berücksichtigt, die Sprechaktivität enthielten. Es wurden also die Silence-Segmente am Anfang und Ende des Satzes entfernt. Zwei geschnittene Sätze von unterschiedlichen Sprechern wurden anschliessend aneinander kopiert und als WAV Datei gespeichert.

Die Ermittlung der Sprechintervalle wurde mit dem gleichen SAD Modell und Einstellungen durchgeführt wie es bereits im Kapitel 5.2.1 erläutert wurde.

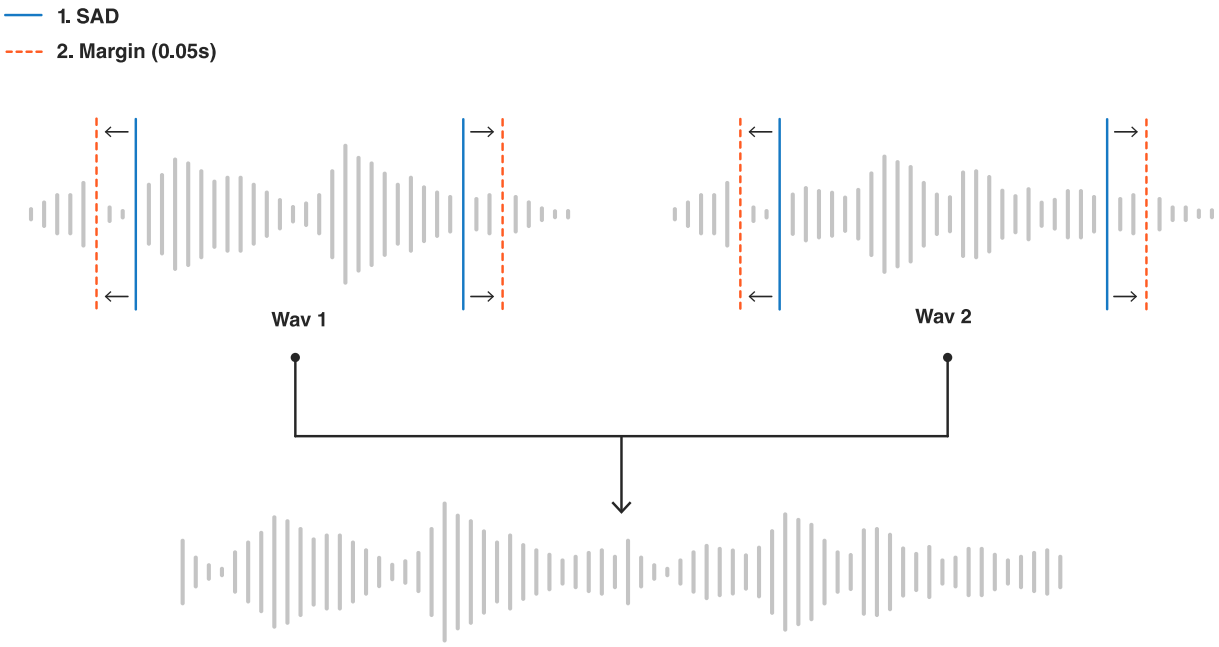


ABBILDUNG 5.5: Visualisierung der Erstellung einer Audiodatei des Szenario 3 Datensatzes.

Insgesamt wurden 2000 solcher Dialoge erzeugt, wobei eine Aufnahme eine Länge von 2 bis 12 Sekunden hat. Die Gesamtlänge an Audio beläuft sich also auf über drei Stunden. Auch hier wurden zusätzlich die Transkriptionen der Sätze gespeichert.

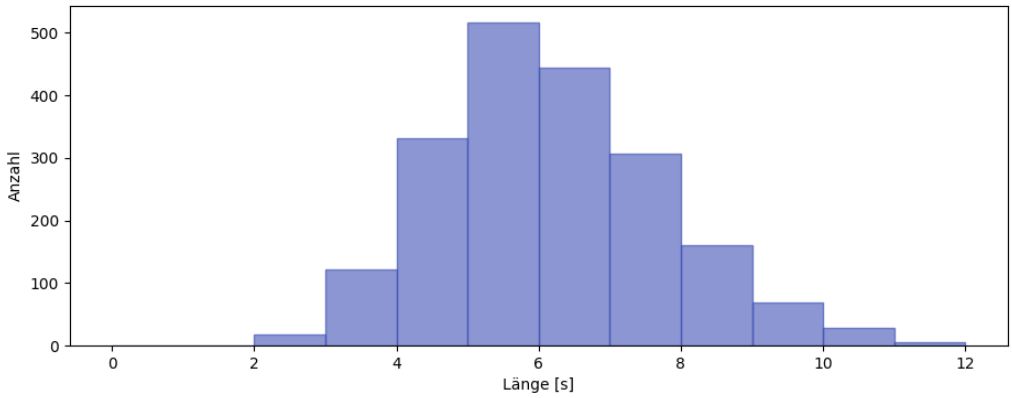


ABBILDUNG 5.6: Histogramm Audiolängen für Szenario 3: Dialog zwischen zwei Personen.

5.3.2 Metadaten

Die Metadaten sind identisch zu Kapitel 5.1.2.

Kapitel 6

Baseline Pipeline

Für diese Arbeit wurde von Dr. Jan Milan Deriu eine Pipeline zur Verfügung gestellt. Die Parameter dieser Pipeline werden mittels Hyperparameter Suche optimiert. Die Pipeline wird in der Arbeit als "Baseline Pipeline" referenziert. Dieses Kapitel beschreibt Aufbau und Funktionsweise der Baseline Pipeline. Da keine offiziellen Auswertungen bezüglich der Performance der Pipeline existieren, wurde auf verschiedenen Korpora eine Evaluierung durchgeführt. Des Weiteren werden verschiedene Analysen präsentiert, mit denen das genaue Verhalten der Pipeline untersucht wurde. Die nachfolgenden Kapitel und Experimente bauen auf den Resultaten dieser Analysen auf.

6.1 Übersicht Baseline Pipeline

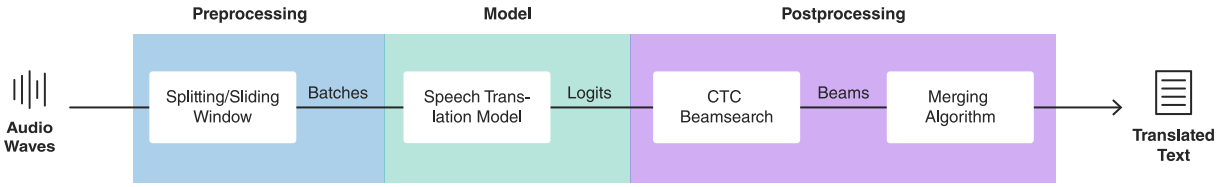


ABBILDUNG 6.1: Übersicht Komponenten der Baseline Pipeline.

6.1.1 Preprocessing

Zum Preprocessing gehören alle Aufgaben, die vor dem ST Modell ausgeführt werden. Das beinhaltet das Einlesen der Audiodatei sowie das Aufsplitten des Audios in einzelne Segmente mittels Sliding Window Verfahren.

Audiodatei Konvertierung

Das ST Modell benötigt Audiodaten mit 16 kHz Abtastrate. Audiodateien werden mittels Librosa geladen und wenn nötig zu 16 kHz konvertiert. Danach erzeugt Librosa ein Numpy Array vom Typ float32, welches dann für die weitere Verarbeitung benutzt wird.

Splitting/Sliding Window

Das ST Modell kann mit genügend Rechenleistung Audiodateien von einer maximalen Länge von 15-20 Sekunden in einem Schritt verarbeiten. Das Modell wurde mit einzelnen Sätzen trainiert, die diese Grenze meist nicht überschreiten. Um längere Sequenzen zu übersetzen, wurde ein Sliding Window Ansatz benutzt. Dabei wird das Audioarray mit Überlappung aufgeteilt und jedes Teilarray gepadded und als Batch zusammengefasst. Der Sliding Window Algorithmus besitzt zwei Hyperparameter, die Windowlänge und die Überlappungsrate.

6.1.2 ST Modell

Das ST Modell besteht aus einem Huggingface Wav2Vec2 Modell, beschrieben in Kapitel 2.8, das zusätzlich noch einen linearen Layer besitzt. Als Output werden Logits erzeugt, die alle 20 ms die Wahrscheinlichkeiten für jeden Token im Vokabular beschreiben. Das Modell ist ein vortrainiertes, multilinguales XLS-R Modell [7] mit einer Milliarde Parametern. Dieses wurde anschliessend auf schweizerdeutschen Daten finegetuned. Die Arbeit basiert auf einem Checkpoint, da das Modell parallel zu dieser Arbeit weitertrainiert wurde.

6.1.3 Postprocessing

Zum Postprocessing gehören alle Aufgaben, die nach dem ST Modell ausgeführt werden. Dazu gehört das CTC Beamsearch sowie das Merging von überlappenden Windowsequenzen.

CTC Beamsearch

Das CTC Beamsearch kombiniert Beam Search 2.7 mit Connectionist Temporal Classification 2.6.

Zusätzlich wird in der Suche ein 5-gram Language Model 2.2 zur Bewertung miteinbezogen. Das verwendete Language Model wurde auf Deutsch mittels Sätzen von Wikipedia trainiert und von Dr. Jan Milan Deriu zur Verfügung gestellt.

Als Eingang bekommt das CTC Beamsearch die Logits vom Modell, die alle 20 ms beschreiben, was die Wahrscheinlichkeit für jeden Token im Vokabular ist. Die Token haben einen Index für jedes Zeichen im Alphabet wie auch weitere Spezialzeichen, die für das CTC gebraucht werden.

Als Tool für das CTC Beamsearch wird pyctcdecode verwendet [44]. In dieser Bibliothek wird der Einfluss des Language Models durch den Hyperparameter alpha angepasst, welcher in dieser Arbeit auf 0.5 gesetzt und nicht verändert wurde. Der Parameter beta beinflusst zudem den Einfluss der Länge auf das Scoring der CTC Beamsearch. Für alpha und beta wurden die Defaultwerte verwendet, da das Finetuning mehrere Wochen in Anspruch genommen hätte.

Als Output liefert die Suche eine Anzahl gefundener Kandidaten. Ein Kandidat besteht dabei aus den einzelnen Wörtern mit Zeitstempeln für Anfang und Ende und den Logit- und LM-Wahrscheinlichkeiten für den kompletten Pfad des Kandidaten. Für die weiteren Berechnungen wurde jeweils nur der beste Kandidat evaluiert.

Merging Algorithmus

Wenn die geladene Audiodatei grösser als die maximal mögliche Länge für das Modell ist, wird vorangehend mittels Sliding Window gesplitted. Anschliessend traversiert jedes Segment das Modell und das CTC Beamsearch. Es resultieren überlappende CTC Beamsearch Kandidaten. Bevor man die Kandidaten verschmelzen kann, muss zuerst der zeitliche Offset des jeweiligen Segmentes bei jedem Wort der Kandidaten addiert werden. Falls in einem zeitlichen Intervall nur ein Wort gefunden wurde, wird dieses übernommen. Gibt es aber Wörter von mehreren Kandidaten, die sich zeitlich überlappen, wird jeweils das Wort ausgewählt, welches am nächsten in der Mitte des Überlappungsintervalles liegt. Die Hypothese dieser Idee war, dass Wörter in der Mitte eines Windows mit grösserer Wahrscheinlichkeit richtig übersetzt werden.

6.2 Hyperparameteroptimierung Baseline Pipeline

Die Baseline Pipeline besitzt einige Parameter, die zuerst mittels Hyperparametersuche optimiert wurden. Folgende Parameter, beschrieben in Kapitel 6.1.1 und 6.1.3, standen für eine Optimierung zur Verfügung:

- lm_alpha
- lm_beta
- n_overlap_sec: Window Size für das Modell
- n_overlap_ratio: Window Overlap in Prozent

6.2.1 Datensatz

Die Hyperparameteroptimierung wurde auf einem separat erzeugten Datensatz durchgeführt. Dafür sollen keine Daten genutzt werden, die in den Auswertungen zu finden sind. Deshalb basiert dieser Datensatz auf der Schweizer Dialektsammlung (siehe Kapitel 4.1), der bereits für das Training des Modells genutzt wurde. Der synthetische Datensatz ist gleich aufgebaut wie der Datensatz für Szenario 1: Lange Audiodatei (siehe Kapitel 5.1). Im Gegensatz zu Szenario 1 wurden allerdings nur 30 Audiodateien generiert, weil ein Suchlauf ansonsten zu lange gedauert hätte.

6.2.2 Optimierung

Es wurde entschieden, dass eine Suche nur für *n_overlap_ratio* und *n_overlap_sec* durchgeführt wird. Der Grund dafür ist zeitlich bedingt, da die Optimierung auf allen Parametern mehrere Wochen in Anspruch genommen hätte. Für *lm_alpha* und *lm_beta* werden die Standardwerte von 0.5 respektive 1 fixiert [44].

Für die Hyperparameter Suche wurde Ray Tune [45] und als Performance Indikator der BLEU Score verwendet. Folgende Werte wurden in einer Gridsuche in Betracht gezogen:

- n_overlap_sec [Sekunden]: 4, 5, 6, 7, 8, 9, 10, 11, 12, 13
- n_overlap_ratio [Prozent]: 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9

6.2.3 Resultat

Folgende Parametereinstellungen erreichten den besten BLEU Score und wurden anschliessend für die Evaluationen verwendet:

- n_overlap_sec [Sekunden]: 10
- n_overlap_ratio [Prozent]: 0.9

Nachfolgend ist der komplette Suchlauf dargestellt. Die Darstellung zeigt für jede Wertekombination den erreichten BLEU Score innerhalb des Datensatzes.

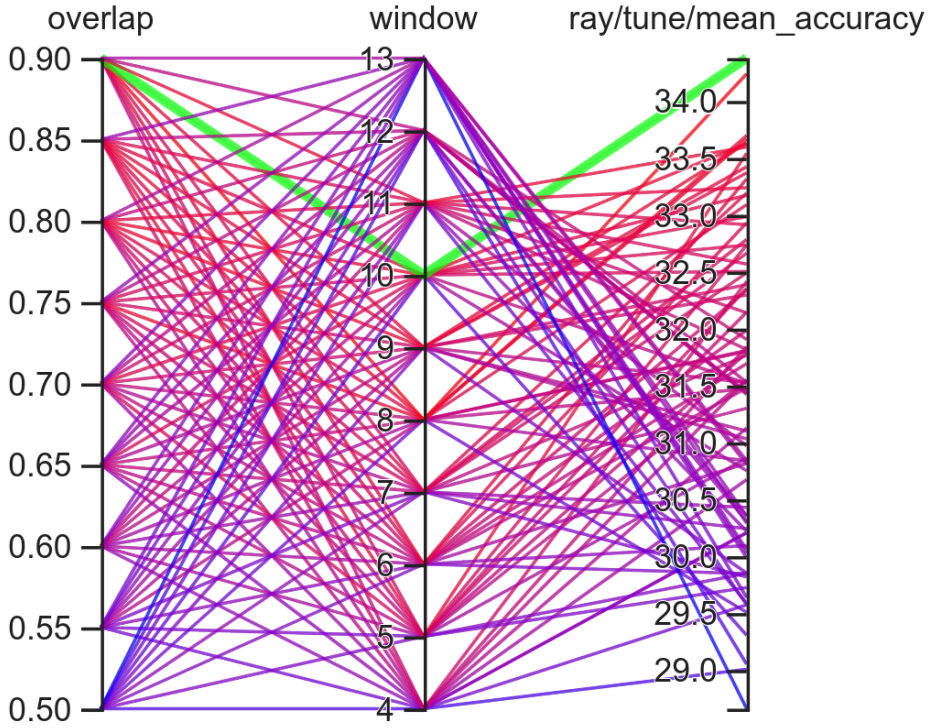


ABBILDUNG 6.2: Resultate Hyperparameter Suche der Baseline Pipeline. Grün dargestellt ist das beste Ergebnis.

6.3 Evaluation Baseline Modell

Die Baseline Pipeline wurde bis jetzt nicht in komplexen Anwendungsfällen getestet sondern nur auf einzelne Sätze mit einer Maximallänge von zehn Sekunden trainiert und evaluiert. In diesem Abschnitt sowie den folgenden Kapiteln werden folgende Datensätze ausgewertet:

- Szenario 1: Lange Audiodatei (Kapitel 5.1)
- Szenario 2: Keine Sprechpausen (Kapitel 5.2)
- Szenario 3: Dialog (Kapitel 5.3)
- SwissText - Einzelne Sätze (Kapitel 4.3)
- Podclub - Monologe von Sprachpodcasts (Kapitel 4.4)

6.3.1 Pipeline Parameter

Folgende Parameter wurden für die Evaluation der Baseline Pipeline genutzt:

- lm_alpha: 0.5
- lm_beta: 1
- n_overlap_sec: 10
- n_overlap_ratio: 0.9
- beam size: 200

6.3.2 Clip Reference

Da bei synthetischen Datensätzen (Szenarios) die Satzgrenzen bekannt sind, werden zudem die Werte gemessen, wenn die Baseline Pipeline jeden Satz einzeln übersetzen würde. Für diesen Zweck wird beim Preprocessing bei den Satzgrenzen ohne Überlappung aufgeteilt. Diese Performance wird für den Rest der Arbeit als ”Clip Reference” oder "CLIP_REFERENCE" definiert. Die Clip Reference ist ein guter Indikator dafür, was die Pipeline maximal leisten könnte. Für nicht synthetisch erzeugte Datensätze kann keine Clip Reference berechnet werden.

6.3.3 Resultate

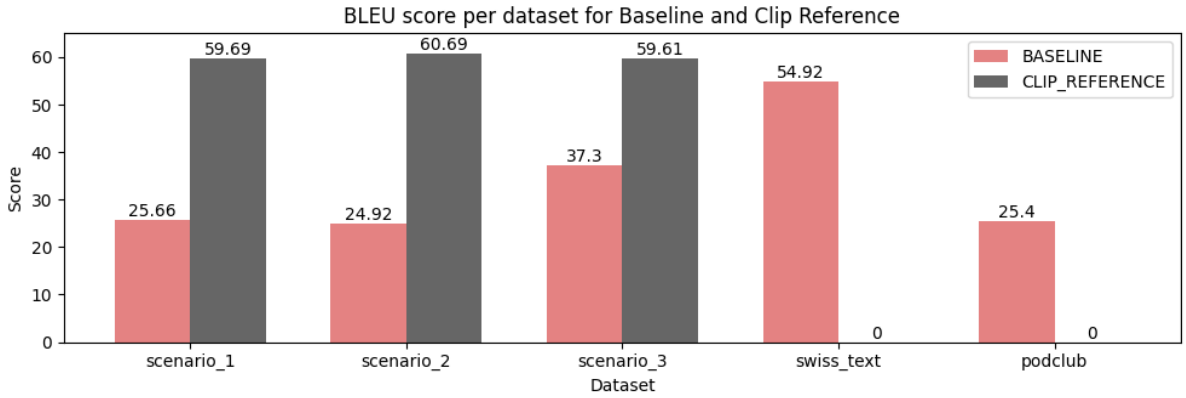


ABBILDUNG 6.3: Vergleich BLEU score zwischen Baseline und Clip Reference. Für SwissText und Podclub existiert keine Clip Reference.

6.3.4 Fazit

Anhand der Clip Reference 6.3.2 in der obenstehenden Abbildung lässt sich aus den drei Szenarios ableiten, dass die einzelnen Sätze zwar sehr gut übersetzt werden könnten, das Aufteilen und Merging jedoch schlecht funktioniert. Im Gegensatz dazu erreicht die Baseline Pipeline einen guten BLEU Score im SwissText Datensatz. Das lässt sich dadurch erklären, dass die meisten Audiodateien nicht länger als die maximale Windowgrösse sind. Das heisst, dass in diesen Fällen kein Splitting und Merging gemacht wird. Ausserdem besteht jede Audiodatei im SwissText Datensatz aus genau einem Satz. Da das ST Modell auf satzweise Übersetzungen trainiert wurde, könnte dies ein weiterer Grund für das gute Ergebnis sein.

6.4 Analyse

In den nachfolgenden Analysen werden Gründe gesucht, wieso die Baseline Pipeline mit dem Sliding Window Ansatz schlechtere Ergebnisse erzielt wie die Clip Reference.

6.4.1 Qualitative Analyse

Qualitative Fehleranalyse

In diesem Abschnitt wird anhand von Beispielen versucht herauszufinden, in welchen Fällen die ST Pipeline Fehler macht. Die Beispiele wurden manuell ausgesucht. Es wird die Ground-Truth, die Clip Reference, sowie die Baseline Ausgabe angegeben. Zudem wird der BLEU Score für die gezeigten Ausschnitte berechnet und dargestellt.

Beispiel 1: Fehlende Punkte (Szenario 1, Index 12)		
Experiment	BLEU Score	Übersetzung
Ground-Truth	-	*Eine Gesamtzahl ist noch nicht bekannt. Bei Menschen mit dunklen Haaren ist das Risiko deutlich geringer. [...]*
Clip Reference	100	*Eine Gesamtzahl ist noch nicht bekannt. Bei Menschen mit dunklen Haaren ist das Risiko deutlich geringer. [...]*
Baseline	78.16	*Eine Gesamtzahl ist noch nicht bekannt Bei Menschen mit dunklen Haaren ist das Risiko deutlich geringer [...]*

TABELLE 6.1: Qualitative Fehleranalyse Beispiel 1: Fehlende Punkte.

Beispiel 1 zeigt, dass die ST Baseline Pipeline in der Lage ist, gute Ergebnisse zu erzielen. Auffällig ist aber, dass die Punkte an den Satzenden fehlen, obwohl diese bei der Clip Reference korrekt gesetzt werden.

Beispiel 2: Referenzübersetzung (SwissText, Index 2874)		
Experiment	BLEU Score	Übersetzung
Ground-Truth	-	*Dieser lehnte die Anfrage jedoch ab.*
Clip Reference	-	-
Baseline	16.52	*Dieser hat jedoch die Anfrage abgelehnt.*

TABELLE 6.2: Qualitative Fehleranalyse Beispiel 2: Referenzübersetzung.

Das Beispiel 2 besitzt genau einen Satz, den die Baseline Pipeline perfekt übersetzt. In diesem Fall wurde auch am Ende der Punkt richtig gesetzt. Der BLEU Score ist jedoch mit 16.52 sehr tief. Das erklärt sich dadurch, dass der Referenzsatz im Präteritum ist.

Weil Schweizerdeutsch kein Präteritum besitzt, spricht der Sprecher den Satz im Perfekt aus. Die Übersetzung wäre somit korrekt, da aber genau 1 Referenzübersetzung im Präteritum hinterlegt wurde, ist der Score so tief. Solche Fälle sind jedoch selten.

Beispiel 3: Komplettes Versagen (Szenario 2, Index 49)		
Experiment	BLEU Score	Übersetzung
Ground-Truth	-	*Wir kommen nun zur Detailberatung. Der Gemeinderat nimmt es auch als Postulat entgegen. Das Budget 2017 wurde mit verschiedenen «Pauschalkürzungen» versehen. [...]*
Clip Reference	45.69	*Wir kommen nun zur Teilberatung. Der Gemeinderat nimmt das alle als Postulat entgegen. Das Budget wurde mit verschiedenen Pauschalkürzungen versehen worden. [...]*
Baseline	4.93	*Wir kommen zu Teilen Gemeine als Postulat gegen Budget <unk><unk><unk>urdmitverschiedenen Pauschalkürzungen versehen Schluss ich verschiedenen Pauschalkürzungen versehen [...]*

TABELLE 6.3: Qualitative Fehleranalyse Beispiel 3: Komplettes Versagen.

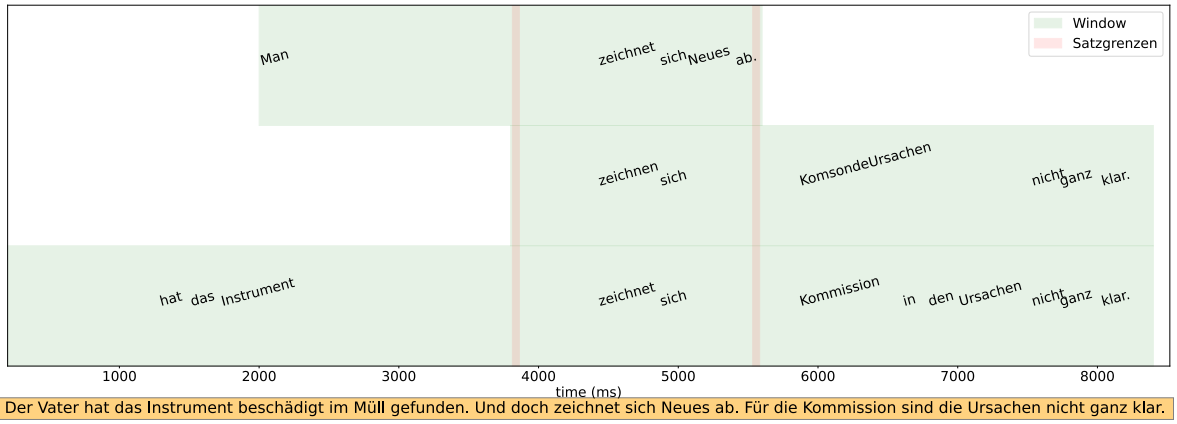
In Beispiel 3 werden einige Probleme ersichtlich. Anhand der Clip Reference ist erkennbar, dass Zahlen ausgelassen werden und auch einige Wörter falsch sind. Die Baseline Pipeline hingegen versagt in diesem Abschnitt komplett. Wörter werden verändert, ausgelassen, zusammenkopiert und kommen plötzlich doppelt vor. Die drei <unk> sind auch unerwartet. Diese kommen vor, wenn das Token nicht im Vokabular gefunden werden kann. Zusätzlich werden auch wieder keine Punkte gesetzt. Die doppelten Wörter werden durch das fehlerhafte Merging erzeugt.

Interpretation Qualitative Fehleranalyse

Mittels Vergleichen von Ground-Truth, Clip Reference und Vorhersage wurden verschiedene Probleme erkannt. Zum einen scheint die Pipeline Schwierigkeiten zu haben, Satzschlusszeichen zu setzen. Beispiel 3 zeigt auf, dass in manchen Fällen die Baseline Pipeline Übersetzungen generiert, die keinen Sinn ergeben, Wörter doppelt einfügt oder sie zusammenkopiert zu einem Wort. Weil diese Probleme in denselben Abschnitten bei der Clip Reference kaum vorkommen, wird im nächsten Abschnitt der Einfluss von Windowgrössen auf die Übersetzung analysiert.

Einfluss von Windowgrössen

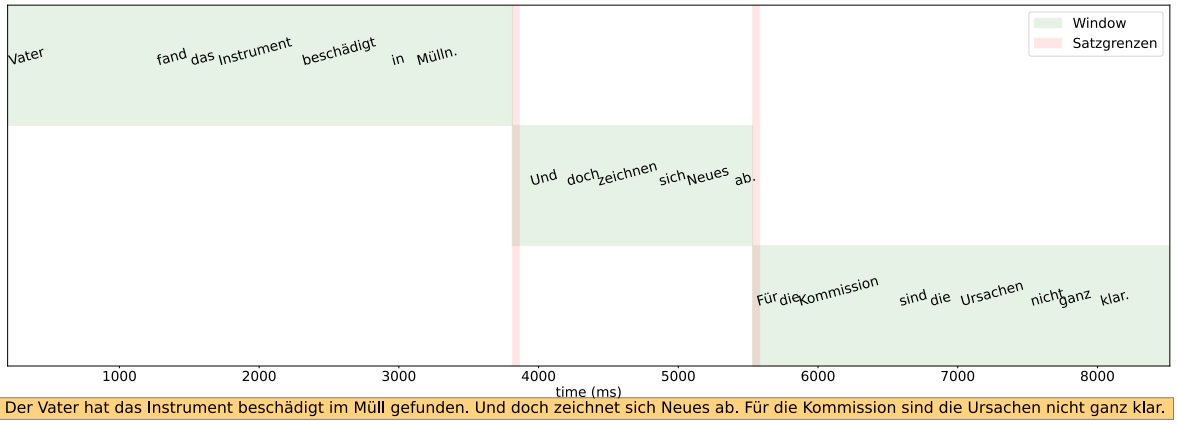
Die unten stehenden Grafiken visualisieren den Output verschiedener Windowgrössen. Es wurden drei Sätze aneinander kopiert. Dieses Audio dient als Input für die nachfolgenden Grafiken. Pro Zeile wird ein Window mit fixer Grösse in Grün dargestellt. In Rot sind die wirklichen Satzgrenzen dargestellt.
Es geht nun darum aufzuzeigen, wie sich der Output anhand verschiedener Windowgrössen und Positionen der Windows verändert. Wie man in den Grafiken erkennen kann, ist die Vorhersage schlecht, wenn das Window nicht genau den Satz trifft. Zum einen fehlen Wörter und es werden Wörter erzeugt, die so nicht existieren.



Ref: Der Vater hat das Instrument beschädigt im Müll gefunden. Und doch zeichnet sich Neues ab. Für die Kommission sind die Ursachen nicht ganz klar.

ABBILDUNG 6.4: Darstellung der Outputs verschiedener Windowgrössen. Die Wörter wurden gedreht für eine bessere Lesbarkeit.

Zum Vergleich ist in untenstehender Abbildung der Output visualisiert, falls die Windows genau die Satzgrenzen treffen (Clip Reference). Es ist klar ersichtlich, dass sich die Übersetzungen stark verbessern, sobald die Satzgrenzen getroffen werden.



Ref: Der Vater hat das Instrument beschädigt im Müll gefunden. Und doch zeichnet sich Neues ab. Für die Kommission sind die Ursachen nicht ganz klar.

ABBILDUNG 6.5: Darstellung der Outputs, wenn die Windowgrössen genau den Satzgrenzen entsprechen. Die Wörter wurden gedreht für eine bessere Lesbarkeit.

6.4.2 Boundary Analyse

Wie die Beispiele aus den vorhergehenden qualitativen Analysen aufzeigen, ist die Baseline selten in der Lage, Satzschlusszeichen korrekt zu setzen. In diesem Kapitel wird dieses Problem mithilfe von Grafiken genauer analysiert.
In der folgenden Grafik wurde für ein Audio, in der ein Satz gesprochen wird, der Logitoutput visualisiert, indem jeweils der wahrscheinlichste Buchstabe pro Logit geplotted wurde. In Gelb steht zudem die Ground-Truth.

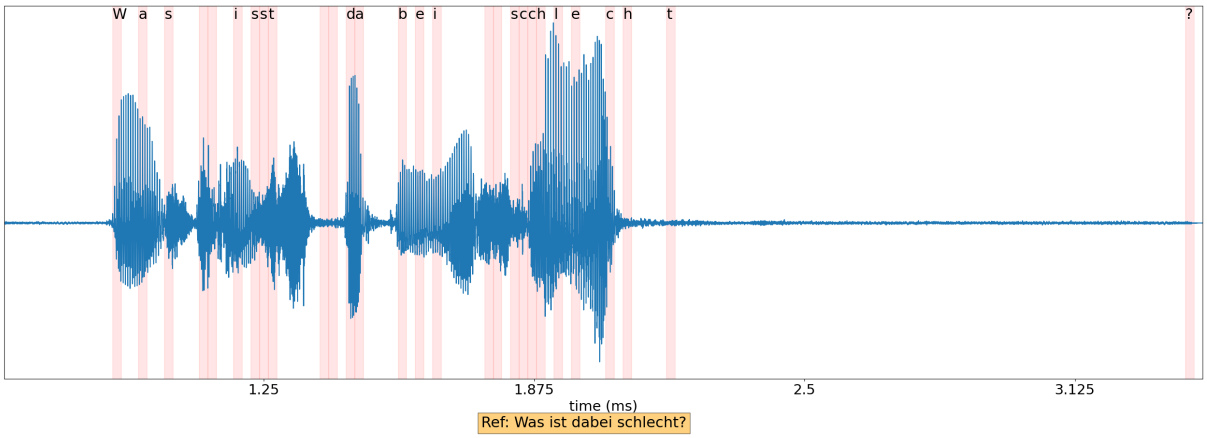


ABBILDUNG 6.6: Output der einzelnen Logits (Buchstaben). Zusätzlich ist die Ground-Truth (Ref) angegeben.

Im obenstehenden Plot ist gut erkennbar, dass Satzschlusszeichen ganz am Ende gemacht werden. In der folgenden Grafik ist der Output nach CTC Beam Search geplotted.

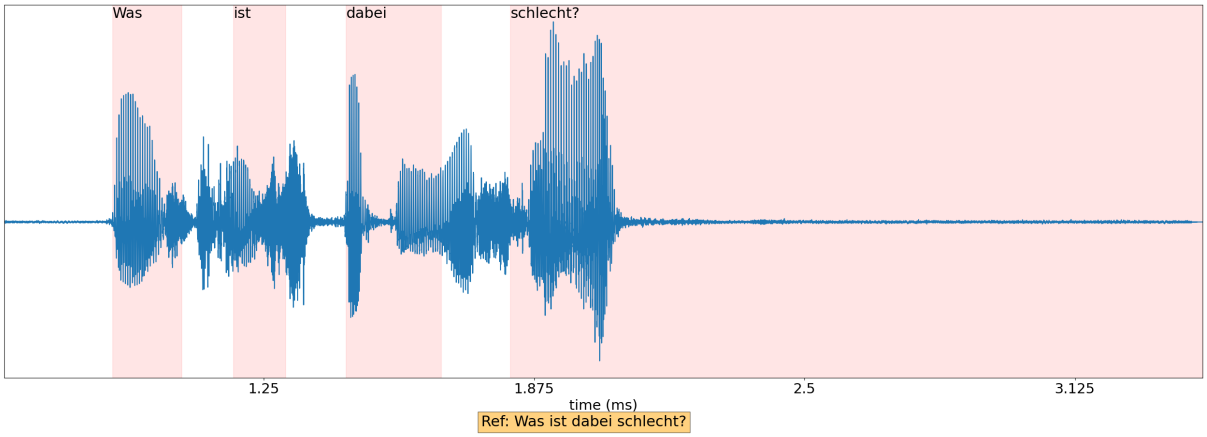


ABBILDUNG 6.7: Output nach der CTC Beam Search (Wörter). Zusätzlich ist die Ground-Truth (Ref) angegeben.

Um das Problem des Satzschlusszeichens noch besser zu veranschaulichen, wurde eine zwanzig Sekunden Audiodatei als Input für das ST Modell verwendet und mittels mehrerer Windowlängen ohne Overlapping ausgewertet.
Die nachfolgende Grafik visualisiert die Wahrscheinlichkeiten der Logits für ein Satzschlusszeichen zur Zeit in Sekunden für unterschiedliche Windowgrössen. Für jede Windowlänge ist die Wahrscheinlichkeit für ein Satzschlusszeichen am Ende des Windows mit Abstand am grössten.

code

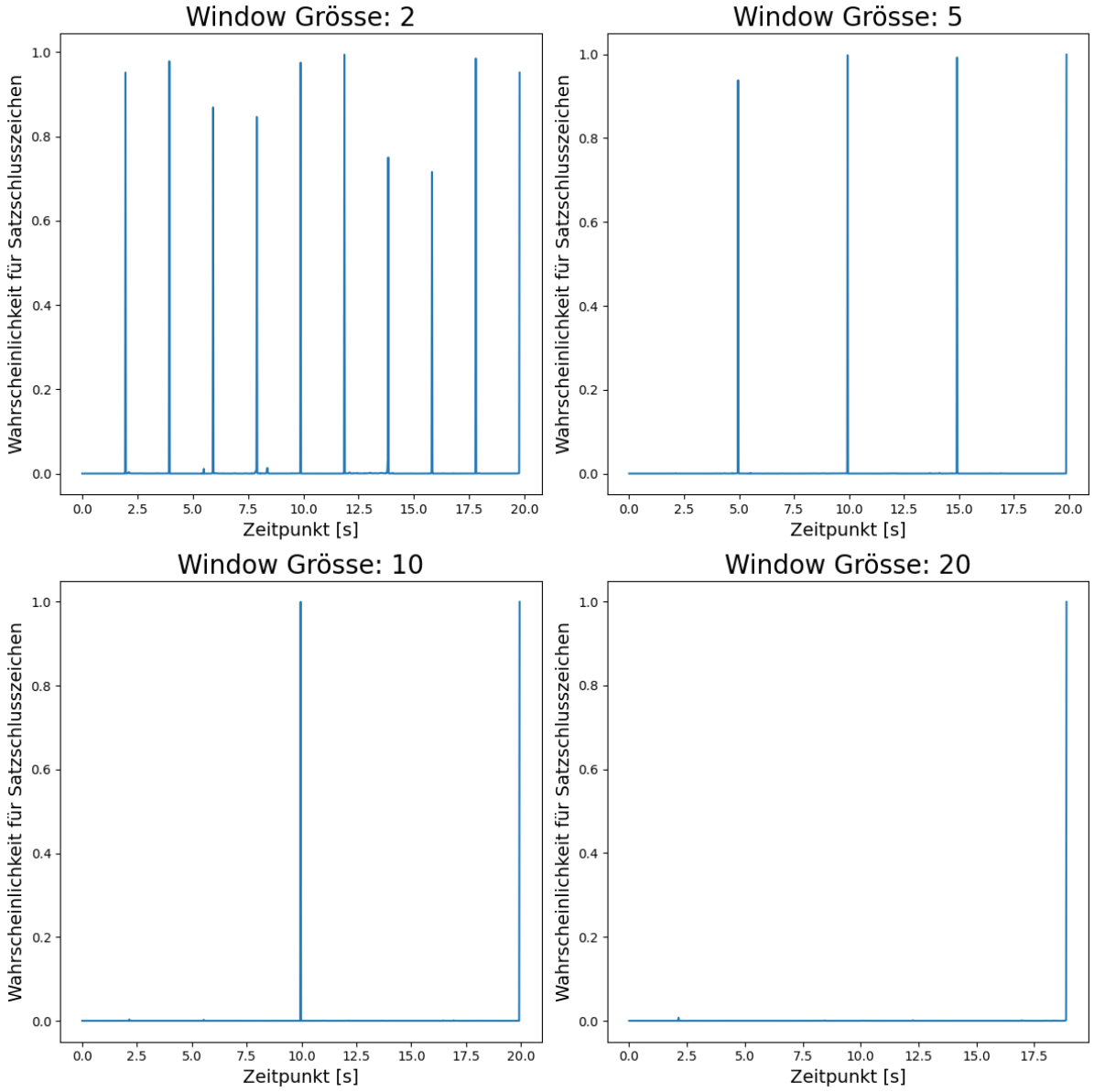


ABBILDUNG 6.8: Wahrscheinlichkeiten für ein Satzschlusszeichen in den Logits über Window Grössen von 2, 5, 10 und 20 Sekunden. Das Audio ist in allen Fällen dasselbe. Overlap und Merging wurde weggelassen.

Ein weiteres Problem ist, dass das Modell das Padding zu ignorieren scheint. Gepadded wird nämlich mit Silence, wodurch das Modell die Satzzeichen ganz zum Schluss des Paddings setzt.

6.5 Fazit

Wie in Abbildung 6.3 erkennbar, könnte das Modell die einzelnen Sätze zufriedenstellend übersetzen (Clip Reference). Besteht eine Eingangssequenz jedoch aus mehreren Sätzen, verschlechtert sich die Vorhersage erheblich. Eines der Probleme ist, dass das Modell für jeden Input ein Satzschlusszeichen macht 6.8. Dieses Verhalten erklärt sich sehr wahrscheinlich dadurch, dass das Modell nur auf einzelnen Sätzen trainiert wurde. Des Weiteren wird sogar das Padding ignoriert und das Satzschlusszeichen hinter das Padding gesetzt, falls man mit Null (Silence) padded. Das ist darum problematisch, da bei einem Batch alle Sequenzen auf die jeweils grösste gepadded werden, damit alle Sequenzen gleich gross sind. Da nun das Padding nicht ignoriert wird und das Satzschlusszeichen am Ende steht, muss das Beamsearch erheblich längere Logitoutputs verarbeiten. Zudem hat die Länge der Sequenz einen Einfluss auf das Beamsearch, wodurch die Berechnung verfälscht wird. Die schlechtere Qualität ist zudem auch dadurch zu erklären, dass vom Schweizerdeutschen ins Hochdeutsche ein Übersetzungsschritt nötig ist, der teilweise auch Satzstrukturen verändert, um im Hochdeutschen die Satzlogik zu gewährleisten. Bekommt das Modell keinen isolierten Satz, ist es viel schwieriger diesen präzise zu übersetzen, wie man in Abbildung 6.4 und 6.5 erkennen kann.

Kapitel 7

Experimente

Basierend auf der Analyse der Baseline Pipeline wurden verschiedene Experimente durchgeführt mit dem Ziel, die Qualität der Übersetzungen zu verbessern. Das Modell 6.1.2 konnte dabei nicht neu- oder weitertrainiert werden und wird deshalb als unveränderbare Konstante betrachtet. Alle Ansätze in der Pipeline, die vor dem Modell sind, werden als Preprocessing bezeichnet und alle Ansätze, die nach dem Modell sind, werden als Postprocessing bezeichnet. Das letzte Experiment mit dem dynamischen Window Ansatz kann dabei keinem zugeordnet werden, da es grundlegend anders funktioniert.

7.1 Preprocessing

Wie Tsiamas et al. [8] und Gaido et al. [11] bereits beschrieben haben, steigert sich die Übersetzungsqualität, wenn die Audiodatei an den Satzgrenzen aufgeteilt wird, bevor es vom ST Modell übersetzt wird. Die Analysen dieser Arbeit, visualisiert in Abbildung 6.4 und 6.5, unterstützen diese Ansicht. Es gibt unterschiedliche Möglichkeiten ein Audio intelligent zu segmentieren. Tsiamas et al. [8] unterscheiden folgende Ansätze:

- fixed-length (Sliding Window)
- pause-based (beispielsweise mit SAD)
- hybrid

Der Sliding Window Ansatz wurde bis anhin verfolgt. Hybride Ansätze [11, 46] erzielen bessere Leistungen als pause-based Ansätze, jedoch wurde kein offizielles Code-Repository dafür gefunden. Der hybride SHAS Ansatz [8] wurde während der Arbeit

veröffentlicht und deshalb nicht evaluiert. Somit wurde sich für Varianten von pause-based Segmentierung entschieden.

7.1.1 Hypothese

Durch intelligentes Aufteilen der Eingabesequenz kann die Übersetzungsqualität im Vergleich zum Sliding Window Ansatz erhöht werden.

7.1.2 Allgemein

In allen Preprocessing-Experimenten wurde der Sliding-Window Ansatz weitgehend durch eine Splitting Strategie ersetzt. Der Sliding-Window Ansatz kommt dadurch nur noch bei Segmenten zum Einsatz, die trotz Aufteilung immer noch länger als die maximale Windowgrösse sind. Segmente die kürzer als die maximale Windowgrösse sind, können dadurch ohne Überlappung übersetzt werden, womit auch kein Merging gemacht werden muss.

7.1.3 SAD

Speech Activity Detection ist im Kapitel 2.4 genauer beschrieben. In den Experimenten wurde als Tool die SAD von pyannote benutzt [40].

7.1.4 Speaker Diarization

Für die Speaker Diarization, beschrieben in Kapitel 2.5, wurden zwei verschiedene Tools verwendet. Einerseits die Anwendung von pyannote [40], wie auch eine interne Rest API der ZHAW.

7.1.5 Pipeline Parameter

Folgende Parameter wurden für die Evaluation der Pipeline mit verschiedenen Preprocessing Massnahmen verwendet:

- lm_alpha: 0.5
- lm_beta: 1
- n_overlap_sec: 10 (Nur relevant, wenn Segment > 10 Sekunden)
- n_overlap_ratio: 0.9 (Nur relevant, wenn Segment > 10 Sekunden)
- beam size: 200

7.1.6 Resultate

Die Pipelines mit den oben beschriebenen Preprocessing-Strategien wurden auf den Datensätzen getestet. Als Referenz wurde jeweils die Performance der Baseline Pipeline und Clip-Referenz visualisiert.

Szenario 1: Längere Audiodatei

Die verschiedenen Pipelines wurden auf dem in Kapitel 5.1 beschriebenen synthetischen Datensatz getestet, welcher aus längeren Audiodateien besteht.

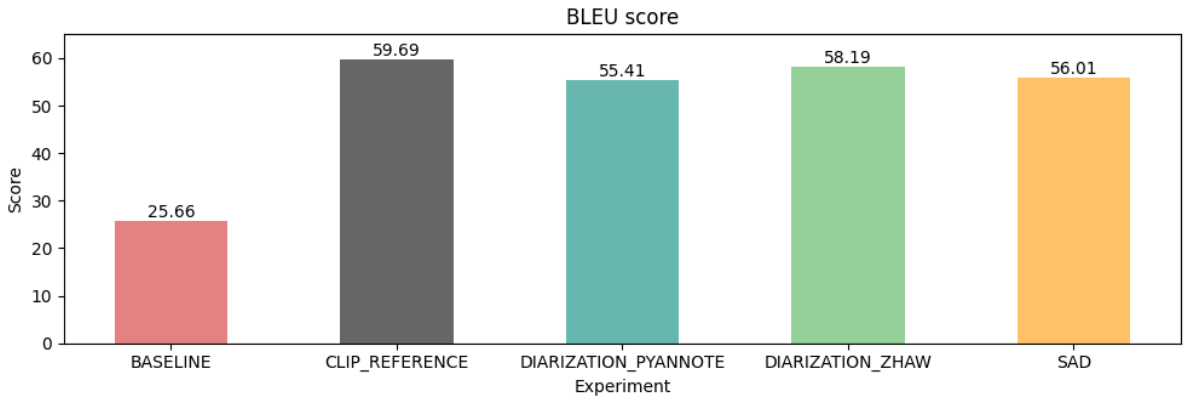


ABBILDUNG 7.1: Resultate Preprocessing auf Szenario 1: Längere Audiodatei

Interpretation

Die verschiedenen Aufteilungsmethoden scheinen einen positiven Effekt auf die Übersetzungsqualität zu haben. Es ist eine erhebliche Steigerung von bis zu 32.53 BLEU Punkten (+126.8 Prozent) gegenüber der Baseline feststellbar. Am besten schneidet die Diarization API der ZHAW ab. Um die Effekte der Aufteilungen besser zu verstehen, wurde gezählt, wie oft eine Methode nicht genau gleich viele Satzgrenzen erkennt wie es die Ground-Truth besitzt. Bei der ZHAW Diarization war dies in 128 der 200 Audiodateien der Fall. Bei SAD in 197 Fällen und bei der Pyannote Diarization in allen der 200 Fällen. Zusätzlich finden SAD und Pyannote Diarization immer mehr Sprechgrenzen als eigentlich vorhanden sind. Bei der ZHAW Diarization ist dies nur in 118 der 128 Fällen der Fall. In 10 Fällen findet die ZHAW Diarization weniger Sprechgrenzen, als in der Ground-Truth. Dies könnte ein möglicher Grund sein, wieso die ZHAW Diarization einen besseren BLEU Score liefert.

Szenario 2: Keine Sprechpausen

Die Pipelines wurden auf dem in Kapitel 5.2 beschriebenen Datensatz evaluiert. Der Datensatz simuliert sehr lange Sprechzeiten ohne Pausen.

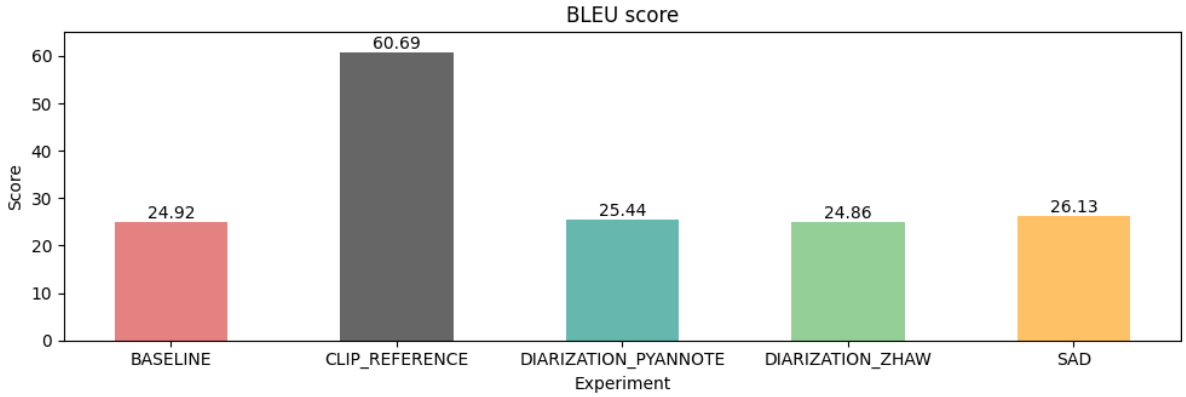


ABBILDUNG 7.2: Resultate Preprocessing auf Szenario 2: Keine Sprechpause

Interpretation

Bei diesem Szenario finden die Varianten keine sinnvollen Satzgrenzen. Die Variante SAD ist knapp einen Punkt besser als die anderen Varianten. Dies liegt aber sehr

wahrscheinlich daran, dass die Audiodaten mit genau demselben SAD Tool zusammengeschnitten wurden und noch ein wenig Toleranz dazu gerechnet wurde (siehe Kapitel 5.2.1). Aus diesem Grund findet SAD in wenigen Audiodateien doch die korrekten Satzgrenzen. Die Pyannote Diarization ist, wie die SAD, ein wenig besser als die Baseline. Dies könnte daran liegen, dass die Pyannote Diarization im Preprocessing SAD einsetzt. Grundsätzlich liefert für das Szenario "Keine Sprechpausen" aber keine der Methoden eine nennenswerte Verbesserung der Übersetzungsqualität.

Szenario 3: Dialog

Die Pipelines wurden auf dem in Kapitel 5.3 beschriebenen Datensatz evaluiert. Der Datensatz simuliert einen Dialog zwischen zwei Personen.

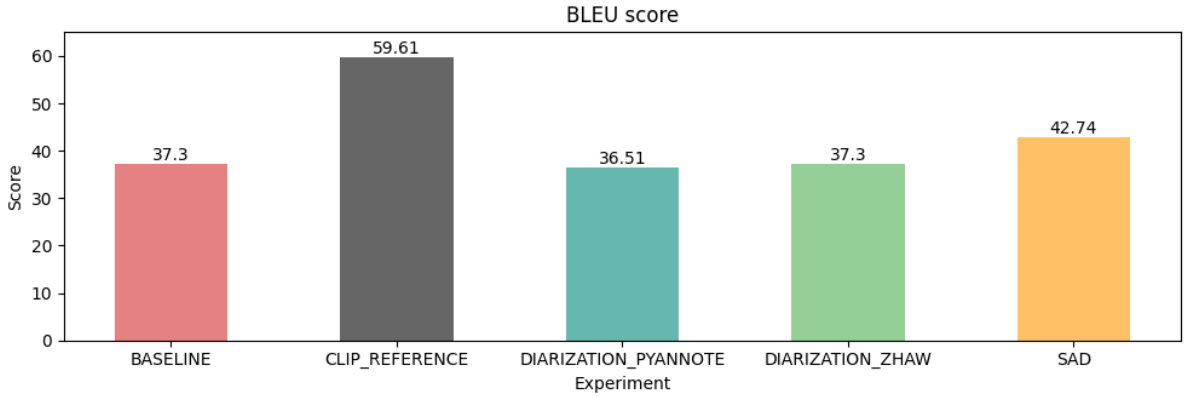


ABBILDUNG 7.3: Resultate Preprocessing auf Szenario 3: Dialog

Interpretation

Auch bei diesem Szenario wurden Satzgrenzen von keiner der Methoden zuverlässig festgestellt. SAD besitzt den besten BLEU Score, vermutlich jedoch aus demselben Grund wie bei Szenario 2: keine Sprechpause. Erstaunlicherweise erkennen beide Diarization Modelle die Satzgrenzen nicht, obwohl nach jedem Satz ein Sprecherwechsel stattfindet. Eine Auswertung, warum Diarization die Sprechergrenzen nicht erkennt, wurde aus Zeitgründen nicht durchgeführt.

SwissText

Die Pipelines wurden auf dem in Kapitel 4.3 beschriebenen Datensatz evaluiert. Der Datensatz besteht aus einzelnen Sätzen.

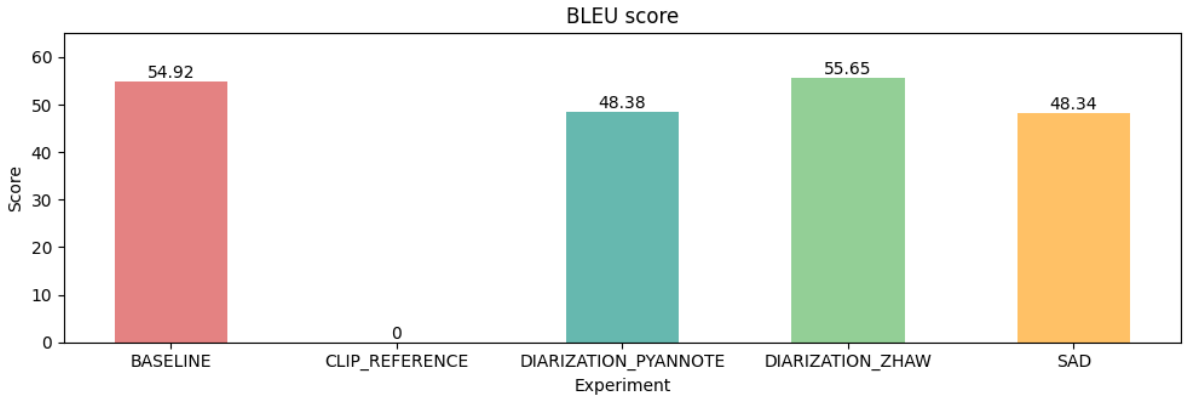


ABBILDUNG 7.4: Resultate auf dem SwissText Datensatz

Interpretation

Auf diesem Datensatz erreicht die ZHAW Diarization einen knapp besseren BLEU Score als die Baseline. Dies kann dadurch erklärt werden, dass Audios, welche über 10 Sekunden lang sind, durch die ZHAW Diarization auf unter 10 Sekunden gekürzt werden, indem Silence entfernt wird. In diesen Fällen ist dann kein Merging nötig und die ZHAW Diarization somit besser als die Baseline Pipeline. SAD und Pyannote Diarization sind schlechter, da sie in über 31 Prozent der Audios sogar mehrere Sprechgrenzen gefunden haben, obwohl nur ein Satz gesprochen wird. Falls mehrere Sprechgrenzen gefunden wurden, ist in 80 Prozent der Fälle auch das Ergebnis schlechter als die Baseline Pipeline. Bei der ZHAW Diarization wurde nur in rund 16 Prozent der Fälle mehrere Sprechgrenzen gefunden. Zudem ist das Ergebnis bei der ZHAW Diarization nur in 50 Prozent der Fälle schlechter, wenn mehrere Sprechgrenzen gefunden wurden.

Podclub

Die Pipelines wurden auf dem in Kapitel 4.4 beschriebenen Datensatz evaluiert. Der Datensatz besteht aus langen Monologen, wobei jeweils eine Geschichte erzählt wird.

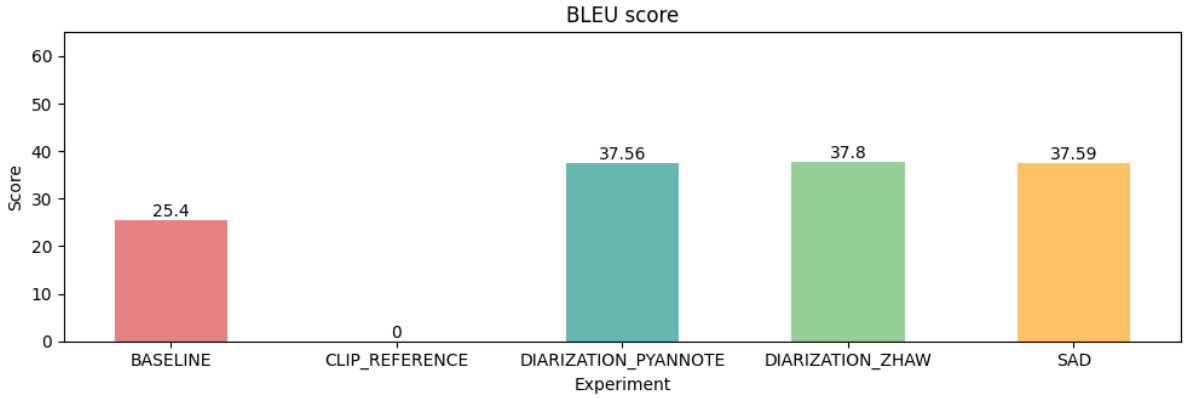


ABBILDUNG 7.5: Resultate auf dem Podclub Datensatz

Interpretation

Auf dem Podclub Datensatz funktionieren alle Preprocessing Methoden ähnlich gut. Alle übertreffen dabei die Baseline um zirka 12 BLEU Punkte (+48 Prozent). Der BLEU Score liegt deutlich unter Szenario 1. Dies zeigt, dass reale Audioaufnahmen durchaus komplexer zu übersetzen sind als einzelne Sätze. Leider steht bei diesem Datensatz keine Clip Reference zur Verfügung. Es wäre interessant zu wissen, wie gut das Modell wäre, wenn es Satz für Satz übersetzen könnte.

7.1.7 Fazit

Die Resultate bestätigen die aufgestellte Hypothese und zeigen auf, dass intelligentes Segmentieren die Übersetzungsqualität der Pipeline in den meisten Fällen verbessert. Auf den künstlich erstellten Datensätzen, die keine Sprechpausen mehr enthalten, funktionieren die gewählten Ansätze aber nicht mehr zuverlässig. Insgesamt liefert die Segmentierung mittels Diarization API der ZHAW die besten Resultate. Die Performance von SAD und die Pyannote Speaker Diarization lässt sich aber steigern, wenn man sie auf schweizerdeutschen Daten trainieren würde. Dafür sind aber Datensätze

nötig, die zusätzlich noch ein Label für Silence besitzen. Es wurde versucht, eine eigene Speaker Diarization basierend auf wav2vec2.0 zu trainieren, dies hat vermutlich wegen dem fehlenden Label für Silence Intervalle nicht funktioniert [47].

7.2 Postprocessing

Die Resultate der Preprocessing Experimente zeigen auf, dass die Eingangssequenz nicht immer in Segmente aufgeteilt werden kann, die kleiner als die maximale Windowgrösse sind. Darum wurde ein Experiment im Bereich Beam Search und Merging Algorithmus durchgeführt, um den Output zu verbessern, wenn gemerged werden muss.

7.2.1 Hypothese

Durch engeres Miteinbeziehen der Beamsearch in den Merging Algorithmus kann die Übersetzungsqualität im Falle von Overlap verbessert werden.

7.2.2 Verbesserter Merging Algorithmus

Wie im Kapitel 6.1.3 beschrieben, liefert das CTC Beamsearch Kandidaten inklusive deren aufmultiplizierten Logit- und LM-Wahrscheinlichkeiten. Die beiden Wahrscheinlichkeiten wurden beim Merging der Baseline Pipeline jedoch nicht in Betracht gezogen. In diesem Experiment wird von den Wörtern, die sich überlappen, jeweils das vom bestbewerteten Pfad ausgewählt.

Hyperparameteroptimierung Verbesserter Merging Algorithmus

Genau gleich wie in Kapitel 6.2 wurde eine Hyperparameter Suche für *n_overlap_ratio* und *n_overlap_sec* durchgeführt. Folgende Werte erreichten den besten BLEU Score und wurden anschliessend für die Evaluationen verwendet:

- n_overlap_sec [Sekunden]: 5
- n_overlap_ratio [Prozent]: 0.75

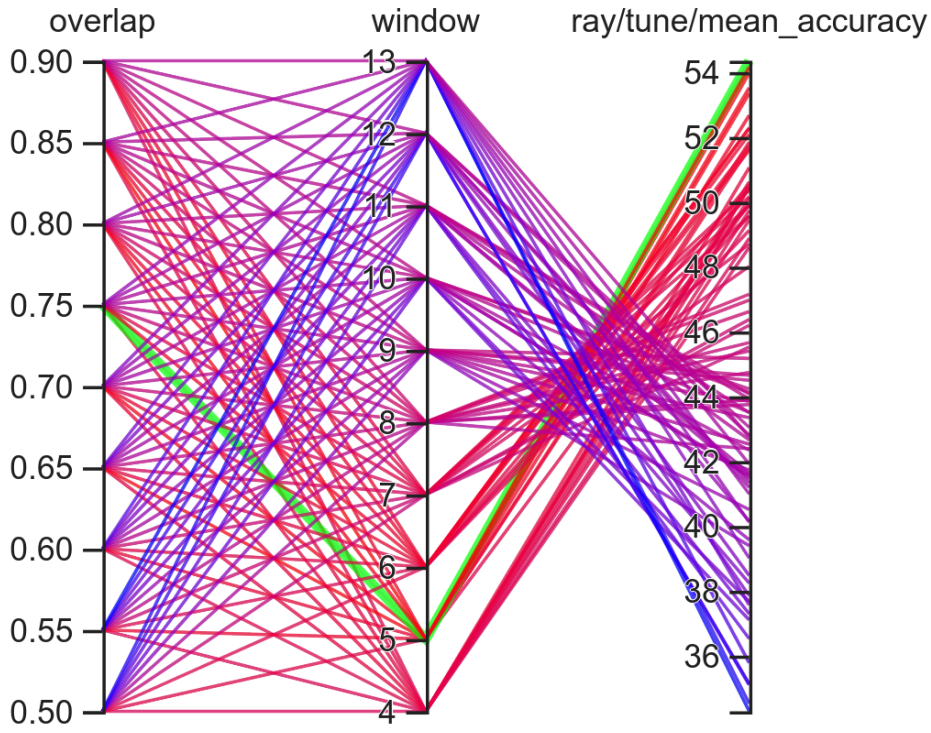


ABBILDUNG 7.6: Resultate Hyperparameter Suche der Pipeline mit Verbessertem Merging Algorithmus. In Grün ist das beste Ergebnis.

7.2.3 Pipeline Parameter

Folgende Parameter wurden für die Evaluation der Pipeline mit verbessertem Merging genutzt:

- lm_alpha: 0.5
- lm_beta: 1
- n_overlap_sec: 5
- n_overlap_ratio: 0.75
- beam size: 200

7.2.4 Resultate

Das Experiment wurde anhand der fünf Korpora ausgewertet. Als Referenz dienen wiederum die Baseline Pipeline, sowie die Clip Referenz.

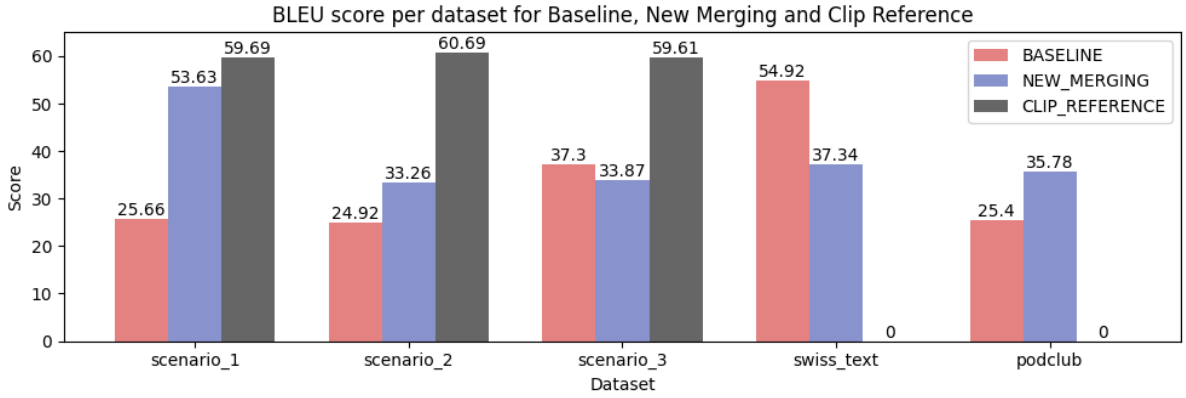


ABBILDUNG 7.7: Vergleich BLEU score zwischen Baseline, Neuem Merging und Clip Reference. Für SwissText und Podclub existiert keine Clip Reference.

7.2.5 Fazit

Das neue Merging verzeichnet auf einigen Datensätzen bessere BLEU Scores als das Baseline Merging. Der enorme Anstieg in Szenario 1 (+109 Prozent BLEU Score) ist

wohl aber darauf zurückzuführen, dass der Datensatz, auf dem die Hyperparametersuche durchgeführt wurde, ähnlich aufgebaut ist wie der von Szenario 1. Das beste Resultat der Suche war eine Windowlänge von 5 Sekunden. Die durchschnittliche Audiolänge im SNF Datensatz, aus dem das Szenario 1 besteht, ist 4.96 Sekunden. Somit bekommt die Pipeline oftmals gute Segmente für die Übersetzung. Dies erklärt auch die viel schlechtere Performance des neuen Merging beim SwissText Datensatz. Dort ist ein Audio durchschnittlich 7.94 Sekunden lang, womit die Pipeline mit dem neuen Merging meist nicht das komplette Audio als Input erhält.

Gerade beim Podclub Datensatz kann das neue Merging bessere Resultate erzielen (+48.9 Prozent BLEU Score). Somit kann die Hypothese aus Kapitel 7.2.1 zwar bestätigt werden, aber die Probleme der Satzgrenzen bleiben bestehen, wenn nur Sliding Window mit neuem Merging eingesetzt wird.

7.3 Dynamic Window Ansatz

Aktuelle Forschungen [8, 11], die eigene Analyse in Kapitel 6.4 sowie die Ergebnisse aus Kapitel 7.1 zeigen, dass sich eine optimale Segmentierung des Audios positiv auf die Übersetzungsqualität auswirkt. Eine weitere Möglichkeit der Segmentierung wäre das Generieren von Übersetzungen via ST Modell und CTC Beamsearch für unterschiedliche Segmentslängen. Von den erzeugten Übersetzungen wird dann mittels Heuristik die beste ausgewählt. Anschliessend startet die Berechnung für das nächste Segment am Endpunkt desjenigen Segments, welches zur besten Übersetzung gehört hat. Dies wird solange wiederholt, bis das Ende der Eingangssequenz erreicht wird.

7.3.1 Hypothese

Die Qualität der Übersetzung kann erhöht werden, indem der Output von dynamischen Windowgrössen evaluiert wird und mithilfe einer Heuristik das beste davon ausgewählt wird.

7.3.2 Überblick

Die Audiodatei wird zuerst von einer Sekunde bis zur maximalen Windowgrösse (10 Sekunden) in Abständen von 250 ms vom Modell ausgewertet. Angehend werden die

Outputs zusätzlich von einem GPT-2 Model bewertet. Anschliessend wird diese Bewertung mit den jeweiligen Logitscores der Kandidaten des CTC Beamsearches multipliziert und das beste Window ausgewählt. Dieser Vorgang wiederholt sich danach jeweils wieder mit Start beim vorherigen Windowende.

Um von allen generierten Outputs der verschiedenen Windowgrössen zwischen 1 bis 10 Sekunden den besten auszuwählen, wurde folgende Heuristik angewendet:

$$sentence\_scores = (softmax(-gpt2\_losses) * 10) \cdot (softmax(logit\_scores) * 10)$$

Die ausgewählte Windowgrösse ist dementsprechend diejenige, welche den höchsten Sentence Score erzielt. Der Score beträgt zwischen 0 und 100, wobei 100 der bestmöglichste Score ist.

Das nachfolgende Bild visualisiert den Dynamic Window Ansatz:

ABBILDUNG 7.8: Visualisierung des Dynamic Window Ansatzes anhand eines Beispielaudios mit zwei Sätzen.

7.3.3 Pipeline Parameter

Folgende Parameter wurden für die Evaluation des Dynamic Window Ansatzes verwendet:

- lm_alpha: 0.5
- lm_beta: 1
- max_window_size: 10 (maximale Windowgrösse pro Satz in Sekunden)
- step_size_ms: 250 (Schrittweite in Millisekunden)
- beam size: 200

Für den Dynamic Window Ansatz werden die Parameter *n_overlap_sec* und *n_overlap_ratio* nicht benötigt, da der Ansatz kein Sliding Window mit Overlap benötigt.

7.3.4 Resultate

Das Experiment wurde an den verschiedenen Korpora ausgewertet. Als Referenz dienen die Baseline Pipeline sowie die Clip Referenz.

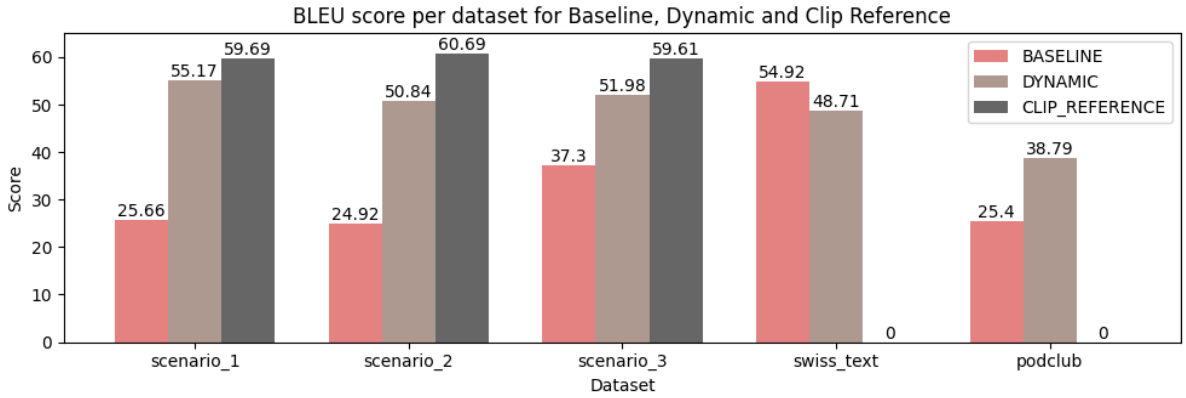


ABBILDUNG 7.9: Vergleich BLEU score zwischen Baseline, Clip Reference und Dynamic Window. Für SwissText und Podclub existiert keine Clip Reference.

7.3.5 Fazit

Der Dynamic Window Ansatz erreicht durchschnittlich gute Ergebnisse. In Szenario 1 mit Sprechpausen ist er zwar schlechter als Preprocessing Ansätze, dafür übertrifft der Ansatz in Szenario 2 und 3 andere Experimente mit einer Steigerung von 52.9 Prozent (Szenario 2 mit neuem Merging), respektive 21.6 Prozent (Szenario 3 mit SAD). Auch im Podclub Datensatz erzielt dieser Ansatz das beste Ergebnis und übertrifft die ZHAW Diarization um knapp 1 BLEU Punkt (+2.6 Prozent). Im SwissText ist der Dynamic Window Ansatz schlechter als die Baseline. Das erklärt sich dadurch, dass die maximale Windowlänge 10 Sekunden beträgt. Im SwissText Datensatz gibt es aber einige Audioaufnahmen, welche länger wie 10 Sekunden sind (siehe Kapitel 4.3). Eine Analyse der Heuristik für die Berechnung des Scores könnte diese Herangehensweise noch verbessern. Dies wurde aus zeitlichen Gründen nicht gemacht.

7.4 Hybrider Ansatz

Experimente im Bereich Preprocessing, Postprocessing sowie der Dynamic Window Ansatz belegen, dass die Leistung der Pipeline stark verbessert werden kann. Nun soll überprüft werden, ob die Leistung weiter erhöht wird, wenn die Verfahren kombiniert werden.

7.4.1 Hypothese

Durch die Kombination einzelner Verfahren kann die Übersetzungsqualität weiter verbessert werden.

7.4.2 Möglichkeiten

Es gibt mehrere Kombinationsmöglichkeiten zwischen den Verfahren. Aus zeitlichen Gründen wurde entschieden, dass nur eine der Kombinationen implementiert wird. Mögliche Kombinationen, die evaluiert wurden sind folgende:

- Kombination 1: Preprocessing Massnahme für Segmentierung und falls Segmente länger als 10 Sekunden sind, wird der Dynamic Window Ansatz verwendet.
- Kombination 2: Preprocessing Massnahme für Segmentierung und falls Segmente länger als 10 Sekunden sind, wird der Sliding Window Ansatz mit neuem Merging verwendet.

Es wird die ZHAW Diarization als Preprocessing verwendet, weil diese auf Szenarien mit Sprechpausen die besten Ergebnisse liefert, siehe Abbildung 7.1. Falls Segmente über 10 Sekunden lang sind, sollen diese mithilfe des Dynamic Window Ansatzes übersetzt werden. Die Überlegung dabei ist folgende: Wenn kurze Segmente gefunden werden, ist die Chance relativ hoch, dass es ein Satz ist. Falls aber ein langes Segment gefunden wird, ist es mit hoher Wahrscheinlichkeit ein Segment mit mehreren Sätzen. Bei Szenarien ohne Sprechpause erzielte der dynamische Window Ansatz die besten Ergebnisse, siehe Abbildung 7.9. Somit wurde für den hybriden Ansatz die Kombination 1 gewählt.

7.4.3 Pipeline Parameter

Folgende Parameter wurden für die Evaluation des hybriden Ansatzes verwendet:

- lm_alpha: 0.5
- lm_beta: 1
- max_window_size: 10
- step_size_ms: 250
- beam size: 200
- max_preprocessing_window: 10

Der Parameter *max_preprocessing_window* ist neu für diesen Ansatz. Er definiert, ab welcher Windowlänge der Dynamic Window Ansatz angewendet werden soll.

7.4.4 Resultate

Der hybride Ansatz wurde an den fünf Korpora ausgewertet. Als Referenz dienen die beiden Ansätze, auf denen der hybride Ansatz aufbaut: die Diarization API, sowie der Dynamic Window Ansatz.

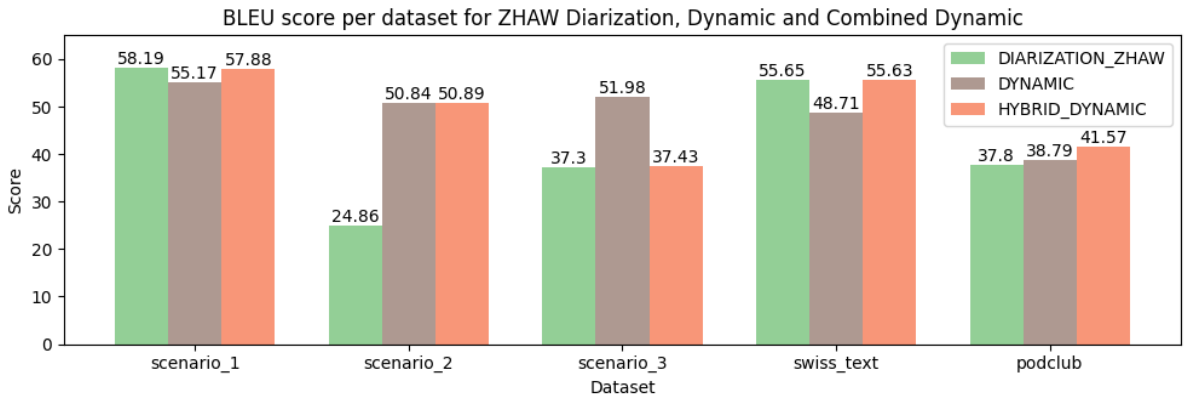


ABBILDUNG 7.10: Vergleich BLEU score zwischen ZHAW Diarization, Dynamic Window und hybrider Ansatz.

7.4.5 Qualitativer Vergleich

In diesem Abschnitt wird der hybride Ansatz qualitativ analysiert und mit den drei Beispielen des Kapitels 6.4.1 verglichen. Damit sollen die Verbesserungen sichtbar gemacht werden.

Beispiel 1: Fehlende Punkte (Szenario 1, Index 12)		
Experiment	BLEU Score	Übersetzung
Ground-Truth	-	*Eine Gesamtzahl ist noch nicht bekannt. Bei Menschen mit dunklen Haaren ist das Risiko deutlich geringer. [...]*
Clip Reference	100	*Eine Gesamtzahl ist noch nicht bekannt. Bei Menschen mit dunklen Haaren ist das Risiko deutlich geringer. [...]*
Baseline	78.16	*Eine Gesamtzahl ist noch nicht bekannt Bei Menschen mit dunklen Haaren ist das Risiko deutlich geringer [...]*
Hybrid Dynamic Window	100	*Eine Gesamtzahl ist noch nicht bekannt. Bei Menschen mit dunklen Haaren ist das Risiko deutlich geringer. [...]*

TABELLE 7.1: Qualitativer Vergleich zwischen Baseline Pipeline und hybrider Ansatz. Beispiel 1: Fehlende Punkte.

In Beispiel 1 werden die Satzschlusszeichen nun korrekt gesetzt. Dies wurde durch die Segmentierung im Preprocessing erreicht. Da die Übersetzung schon sehr gut funktioniert und auch die Punkte am Ende gesetzt werden, ist der BLEU Score in diesem Abschnitt bei 100.

Beispiel 2: Referenzübersetzung (SwissText, Index 2874)		
Experiment	BLEU Score	Übersetzung
Ground-Truth	-	*Dieser lehnte die Anfrage jedoch ab.*
Clip Reference	-	-
Baseline	16.52	*Dieser hat jedoch die Anfrage abgelehnt.*
Hybrid Dynamic Window	16.52	*Dieser hat jedoch die Anfrage abgelehnt.*

TABELLE 7.2: Qualitativer Vergleich zwischen Baseline Pipeline und hybrider Ansatz. Beispiel 2: Referenzübersetzung.

Die Übersetzung ändert sich in Beispiel 2 nicht. Das Problem der Referenzübersetzung im Präteritum bleibt bestehen.

Beispiel 3: Komplettes versagen (Szenario 2, Index 49)		
Experiment	BLEU Score	Übersetzung
Ground-Truth	-	*Wir kommen nun zur Detailberatung. Der Gemeinderat nimmt es auch als Postulat entgegen. Das Budget 2017 wurde mit verschiedenen «Pauschalkürzungen» versehen. [...]*
Clip Reference	45.69	*Wir kommen nun zur Teilberatung. Der Gemeinderat nimmt das alle als Postulat entgegen. Das Budget wurde mit verschiedenen Pauschalkürzungen versehen worden. [...]*
Baseline	4.93	*Wir kommen zu Teilen Gemeine als Postulat gegen Budget <unk><unk><unk>urdmitverschiedenen Pauschalkürzungen versehen Schluss ich verschiedenen Pauschalkürzungen versehen [...]*
Hybrid Dynamic Window	32.24	*Wir kommen nun zur Teilberatung. Der Gemeinderat nimmt es an. Budget wurde mit verschiedenen Pauschalkürzungen versehen. [...]*

TABELLE 7.3: Qualitativer Vergleich zwischen Baseline Pipeline und hybrider Ansatz. Beispiel 3: Komplettes versagen.

Das Beispiel 3 zeigt, dass die Übersetzungsqualität verbessert wurde und näher bei der Clip Reference liegt. Trotzdem werden immer noch Wörter und Satzteile ausgelassen oder verändert. In diesem Beispiel sind die berechneten Intervalle der Segmentierung mithilfe der ZHAW Diarization über zehn Sekunden lang. Somit wurde der dynamische Ansatz verwendet, um die Satzgrenzen zu berechnen. Das lässt darauf schliessen, dass die Heuristik des Dynamischen Ansatzes noch verbessert werden sollte.

7.4.6 Fazit

Die Kombination zwischen Preprocessing und Dynamic Window Ansatz liefert auf einigen Datensätzen die besten Resultate und bestätigt somit die Hypothese. Einzig in Szenario 3 konnte der hybride Ansatz keine guten Resultate erzielen, was dadurch erklärt werden kann, dass es in den 2000 Audiodaten nur 30 Mal vorkommt, dass ein Segment über 10 Sekunden lang ist. Folglich wurden nur die Segmentierungen der Diarization in Betracht gezogen. Das erklärt auch, weshalb der Score fast gleich ist,

wie wenn nur Speaker Diarization im Einsatz ist. Erfreulich ist der Anstieg des BLEU Scores im Podclub Datensatz um ganze 2.78 Punkte (+7.2 Prozent) gegenüber des dynamischen Ansatzes, der zuvor die beste Übersetzungsqualität generiert hat. Die maximale Länge in Sekunden, ab wann ein Segment mittels dynamic Ansatz bewertet wird, könnte via Hyperparametersuche getuned werden. Aus zeitlichen Gründen konnte dies aber nicht gemacht werden.

Der qualitative Vergleich zeigt deutliche Verbesserungen in der Übersetzungsqualität im Vergleich zur Baseline Pipeline. Trotzdem ist die Übersetzung nicht immer verständlich und teilweise nutzlos.

Kapitel 8

Resultate

Für eine umfassende Übersicht werden auf der nächsten Seite alle BLEU Scores sowie Word Error Rates in einer Tabelle dargestellt. Dabei wird für die Clip Reference Blau verwendet, um hervorzuheben, dass diese in realen Bedingungen nicht vorhanden ist. Pro Datensatz wird das beste Ergebnis in **Fett** dargestellt (Ausnahme: Clip Reference).

Für das Verständnis werden die Datensätze nochmals kurz beschrieben:

- Szenario 1: Lange Audiodatei (Kapitel 5.1)
- Szenario 2: Keine Sprechpausen (Kapitel 5.2)
- Szenario 3: Dialog (Kapitel 5.3)
- SwissText - Einzelne Sätze (Kapitel 4.3)
- Podclub - Monologe von Sprachpodcasts (Kapitel 4.4)

BLEU score					
Experiment	SwissText	Podclub	Szenario 1	Szenario 2	Szenario 3
CLIP_REFERENCE	-	-	59.69	60.69	59.61
Baseline	54.92	25.4	25.66	24.92	37.3
SAD	48.34	37.59	56.01	26.13	42.74
Pyannote Diarization	48.38	37.56	55.41	25.44	36.51
ZHAW Diarization	**55.65**	37.8	**58.19**	24.86	37.3
New Merging	37.34	35.78	53.63	33.26	33.87
Dynamic Window	48.71	38.79	55.17	50.84	**51.98**
Hybrid Dynamic Window	55.63	**41.57**	57.88	**50.89**	37.43

TABELLE 8.1: Komplette Übersicht BLEU Scores pro Experiment und Datensatz.

Word Error Rates (WER)					
Experiment	SwissText	Podclub	Szenario 1	Szenario 2	Szenario 3
CLIP_REFERENCE	-	-	0.27	0.27	0.27
Baseline	0.34	0.65	0.63	0.67	0.43
SAD	0.38	0.47	0.32	0.62	0.4
Pyannote Diarization	0.38	0.47	0.33	0.62	0.48
ZHAW Diarization	0.32	0.45	0.29	0.69	0.43
New Merging	0.51	0.47	0.34	0.45	0.46
Dynamic Window	0.4	0.47	0.33	**0.36**	**0.36**
Hybrid Dynamic Window	**0.31**	**0.42**	**0.28**	**0.36**	0.43

TABELLE 8.2: Komplette Übersicht WER pro Experiment und Datensatz.

Kapitel 9

Diskussion

Im Rahmen dieser Bachelorarbeit konnten wir zeigen, dass die Übersetzungsqualität für längere Audiodateien stark erhöht werden konnte, ohne dabei das eigentliche ST Modell zu verändern. Durch synthetische Erstellung von Datensätzen konnten verschiedene Szenarien simuliert werden. Anhand dieser Szenarien konnte dargelegt werden, dass die Baseline Pipeline unbefriedigende Ergebnisse erzielt, sobald mehrere Sätze zusammen übersetzt werden müssen. Es wurden verschiedene Experimente anhand dieser Szenarien evaluiert. Der Ansatz, der dabei die besten Ergebnisse erzielt hat, ist die Kombination aus einer Speaker Diarization der ZHAW und dem dynamischen Window Ansatz aus Kapitel 7.4. Auf dem Podclub Datensatz, der aus Monologen mit einer Dauer von durchschnittlich zehn Minuten besteht, konnte der BLEU Score um 64 Prozent auf 41.57 Punkte verbessert werden. Damit haben wir unser Ziel erreicht, die bestehende Pipeline zu verbessern. Anhand der Clip-Reference der Szenarios ist jedoch gut erkennbar, dass die Ansätze bei weitem noch nicht theoretisch mögliche BLEU Scores erreichen. Dies lässt uns zum Schluss kommen, dass für eine optimale Übersetzung die korrekte Segmentierung des Audios nach einzelnen Sätzen essentiell ist - zumindest was den heutigen Stand der Forschung anbelangt. Betreffend allgemeiner Übersetzungsqualität des besten Kandidaten muss leider die Aussage gemacht werden, dass unser System allgemeine Aufgaben wie die Protokollierung von Meetings, Erstellung von Untertiteln für Filme nicht befriedigend erledigen kann. Neben den Beispielen in der Arbeit finden sich auch noch Übersetzungen im Anhang A.1.

Bei den erreichten Werten ist zu berücksichtigen, dass in allen Ansätzen eine fixe Beam Size von 200 verwendet wurde, weil die Kombination aller Experimente mit verschiedenen Beam Sizes zu viel Zeit in Anspruch genommen hätte. Mit einer grösseren Beam Size würde die Übersetzungsqualität mit Sicherheit noch einmal gesteigert werden können. Weiter wurde in dieser Arbeit die Laufzeit der Experimente nicht gemessen,

da der Fokus einzig und allein auf den Ergebnissen lag. Gerade der dynamische Window Ansatz, der für jeden Satz verschiedene Windowgrössen gleichzeitig berechnet, hat dadurch eine lange Verarbeitungszeit. Des Weiteren ist uns gegen Ende der Arbeit aufgefallen, dass die Erstellung der synthetischen Datensätze für Szenario 2 und 3 fehlerhaft ist, da in seltenen Fällen Sätze im Audio abgeschnitten werden, weil die SAD diese nicht als Sprechaktivität erkannt hat. In diesen Fällen stimmt das Audio nicht mit der Referenzübersetzung überein. Dieser Fehler wurde zu spät entdeckt, weshalb die zeitintensiven Experimente nicht wiederholt werden konnten.

Für zukünftige Arbeiten wäre es interessant, den Ansatz der Boundary Detection aus der Arbeit von Tsiamas et al. [8] mit schweizerdeutschen Daten zu trainieren und in der Folge zu testen. Eine weitere Frage ist, ob ein ST Modell Satzgrenzen selber erkennen kann, falls es auch mit Inputs aus mehreren Sätzen trainiert würde. Ausserdem würden wir gerne eine detaillierte Analyse und Verbesserung der Heuristik des dynamischen Window Ansatzes durchführen. Schliesslich ist uns aufgefallen, dass die Länge des Padding einen Einfluss auf das CTC Beamsearch zu haben scheint. In der Analyse in Kapitel 6.4.2 haben wir bereits beschrieben, dass das Satzzeichen jeweils am Schluss des Paddings gemacht wird. Wir sind aber fälschlicherweise nicht davon ausgegangen, dass das Padding auch in dieser Beziehung Einfluss auf das Ergebnis des Beamsearches hat. Auch dieser Umstand sollte in einer nachfolgenden Arbeit genauer analysiert werden.

Akronyme

ASR Automatic Speech Recognition.

AST Automatic Speech Translation.

AUC Area Under the ROC Curve.

BiLSTM Bidirectional Long Short-Term Memory.

BLEU Bilingual Evaluation Understudy.

CNN Convolutional Neural Network.

CTC Connectionist Temporal Classification.

DER Diarization Error Rate.

DNN Deep Neural Network.

FPR False Positive Rate.

GPU Graphics Processing Unit.

LM Language Model.

NLP Natural Language Processing.

ROC Receiver Operating Characteristic.

SAD Speech Activity Detection.

ST Speech Translation.

STT Speech To Text.

TPR True Positive Rate.

VAD Voice Activity Detection.

WER Word Error Rate.

Literatur

[1] *Speech Translation | Microsoft Azure*. URL: https://azure.microsoft.com/en-in/services/cognitive-services/speech-translation/#overview.

[2] *How Alexa's new Live Translation for conversations works - Amazon Science*. URL: https://www.amazon.science/blog/how-alexas-new-live-translation-for-conversations-works.

[3] *Translate by speech - Computer - Google Translate Help*. URL: https://support.google.com/translate/answer/6142468?hl=en-GB&co=GENIE.Platform%3DDesktop#.

[4] Michel Plüss, Lukas Neukom und Manfred Vogel. *SwissText 2021 Task 3: Swiss German Speech to Standard German Text*. URL: http://ceur-ws.org/Vol-2957/sg_paper1.pdf.

[5] Yuriy Arabskyy et al. „Dialectal Speech Recognition and Translation of Swiss German Speech to Standard German Text: Microsoft's Submission to SwissText 2021". In: *arXiv:2106.08126 [cs, eess]* (Juli 2021). arXiv: 2106.08126. URL: http://arxiv.org/abs/2106.08126 (besucht am 18. 04. 2022).

[6] Michel Plüss et al. *SDS-200: A Swiss German Speech to Standard German Text Corpus*. 2022. DOI: 10.48550/ARXIV.2205.09501. URL: https://arxiv.org/abs/2205.09501.

[7] Arun Babu et al. „XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale". In: *arXiv:2111.09296 [cs, eess]* (Dez. 2021). arXiv: 2111.09296. URL: http://arxiv.org/abs/2111.09296 (besucht am 18. 04. 2022).

[8] Ioannis Tsiamas et al. *SHAS: Approaching optimal Segmentation for End-to-End Speech Translation*. 2022. DOI: 10.48550/ARXIV.2202.04774. URL: https://arxiv.org/abs/2202.04774.

[9] Tirza Biron et al. „Automatic detection of prosodic boundaries in spontaneous speech". In: *PLOS ONE* 16.5 (Mai 2021). Hrsg. von Claudia Männel, e0250969. DOI: 10.1371/journal.pone.0250969. URL: https://doi.org/10.1371/journal.pone.0250969.

[10] Andreas Tsiartas et al. „Robust word boundary detection in spontaneous speech using acoustic and lexical cues“. In: *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. 2009, S. 4785–4788. DOI: 10.1109/ICASSP.2009.4960701.

[11] Marco Gaido et al. *Beyond Voice Activity Detection: Hybrid Audio Segmentation for Direct Speech Translation*. 2021. DOI: 10.48550/ARXIV.2104.11710. URL: https://arxiv.org/abs/2104.11710.

[12] Alexei Baevski et al. „wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations“. In: *arXiv:2006.11477 [cs, eess]* (Okt. 2020). arXiv: 2006.11477. URL: http://arxiv.org/abs/2006.11477 (besucht am 18. 04. 2022).

[13] Matthias Sperber und Matthias Paulik. *Speech Translation and the End-to-End Promise: Taking Stock of Where We Are*. 2020. DOI: 10.48550/ARXIV.2004.06358. URL: https://arxiv.org/abs/2004.06358.

[14] Jurafsky Daniel und James H Martin. „Speech and Language Processing“. In: (2021).

[15] Alec Radford et al. „Language Models are Unsupervised Multitask Learners“. In: 2019.

[16] Tom B. Brown et al. *Language Models are Few-Shot Learners*. 2020. DOI: 10.48550/ARXIV.2005.14165. URL: https://arxiv.org/abs/2005.14165.

[17] *OpenAI API*. URL: https://openai.com/api/.

[18] *dbmdz/german-gpt2 · Hugging Face*. URL: https://huggingface.co/dbmdz/german-gpt2.

[19] Irene Rivera-Trigueros. „Machine translation systems and quality assessment: a systematic review“. In: *Language Resources and Evaluation* (Apr. 2021). ISSN: 1574-0218. DOI: 10.1007/s10579-021-09537-5. URL: https://doi.org/10.1007/s10579-021-09537-5.

[20] Kishore Papineni et al. „Bleu: a Method for Automatic Evaluation of Machine Translation“. In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, Juli 2002, S. 311–318. DOI: 10.3115/1073083.1073135. URL: https://aclanthology.org/P02-1040.

[21] Alon Lavie und Abhaya Agarwal. „METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments“. In: (Juli 2007), S. 228–231.

[22] *Modelle bewerten | AutoML Translation-Dokumentation | Google Cloud*. URL: https://cloud.google.com/translate/automl/docs/evaluate?hl=de#bleu.

[23] Tae Jin Park et al. *A Review of Speaker Diarization: Recent Advances with Deep Learning*. 2021. arXiv: 2101.09624 [eess.AS].

[24] Amit Sofer und Shlomo E. Chazan. *CNN self-attention voice activity detector*. 2022. DOI: 10.48550/ARXIV.2203.02944. URL: https://arxiv.org/abs/2203.02944.

[25] Nicholas Wilkinson und Thomas Niesler. *A Hybrid CNN-BiLSTM Voice Activity Detector*. 2021. DOI: 10.48550/ARXIV.2103.03529. URL: https://arxiv.org/abs/2103.03529.

[26] Yann LeCun und Yoshua Bengio. „Convolutional Networks for Images, Speech, and Time Series“. In: *The Handbook of Brain Theory and Neural Networks*. Cambridge, MA, USA: MIT Press, 1998, S. 255–258. ISBN: 0262511029.

[27] Alex Graves und Jürgen Schmidhuber. „Framewise phoneme classification with bidirectional LSTM and other neural network architectures“. In: *Neural Networks* 18.5 (2005). IJCNN 2005, S. 602–610. ISSN: 0893-6080. DOI: https://doi.org/10.1016/j.neunet.2005.06.042. URL: https://www.sciencedirect.com/science/article/pii/S0893608005001206.

[28] Francisco Melo. „Area under the ROC Curve“. In: *Encyclopedia of Systems Biology*. Hrsg. von Werner Dubitzky et al. New York, NY: Springer New York, 2013, S. 38–39. ISBN: 978-1-4419-9863-7. DOI: 10.1007/978-1-4419-9863-7_209. URL: https://doi.org/10.1007/978-1-4419-9863-7_209.

[29] *By cmglee, MartinThoma - Roc-draft-xkcd-style.svg, CC BY-SA 4.0*. URL: https://commons.wikimedia.org/w/index.php?curid=109730045.

[30] Aonan Zhang et al. „Fully supervised speaker diarization“. In: *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2019, S. 6301–6305.

[31] Hervé Bredin et al. *pyannote.audio: neural building blocks for speaker diarization*. 2019. DOI: 10.48550/ARXIV.1911.01255. URL: https://arxiv.org/abs/1911.01255.

[32] Alexandra Canavan, David Graff und George Zipperlen. *CALLHOME American English Speech LDC97S42*. 1997. DOI: https://doi.org/10.35111/exq3-x930. URL: https://catalog.ldc.upenn.edu/LDC97S42.

[33] Alex Graves et al. „Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural 'networks“. In: Bd. 2006. Jan. 2006, S. 369–376. DOI: 10.1145/1143844.1143891.

[34] *9.6. Encoder-Decoder Architecture — Dive into Deep Learning 0.17.4 documentation*. URL: https://d2l.ai/chapter_recurrent-modern/beam-search.html.

[35] Thomas Wolf et al. „Transformers: State-of-the-Art Natural Language Processing“. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Okt. 2020, S. 38–45. URL: https://www.aclweb.org/anthology/2020.emnlp-demos.6.

[36] *HuggingFace*. URL: https://huggingface.co.

[37] Brian McFee et al. *librosa/librosa: 0.9.1*. Version 0.9.1. Feb. 2022. DOI: 10.5281/zenodo.6097378. URL: https://doi.org/10.5281/zenodo.6097378.

[38] Matt Post. „A Call for Clarity in Reporting BLEU Scores“. In: *Proceedings of the Third Conference on Machine Translation: Research Papers*. Belgium, Brussels: Association for Computational Linguistics, Okt. 2018, S. 186–191. URL: https://www.aclweb.org/anthology/W18-6319.

[39] *JiWER: Similarity measures for automatic speech recognition evaluation*. URL: https://github.com/jitsi/jiwer.

[40] *GitHub - pyannote/pyannote-audio: Neural building blocks for speaker diarization: speech activity detection, speaker change detection, overlapped speech detection, speaker embedding*. URL: https://github.com/pyannote/pyannote-audio.

[41] *Openstack Cluster ZHAW*. URL: https://apu.cloudlab.zhaw.ch/.

[42] *NVIDIA T4 TENSOR CORE GPU*. URL: https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/tesla-t4/t4-tensor-core-datasheet-951643.pdf.

[43] *Schweizer Dialektsammlung*. URL: https://dialektsammlung.ch.

[44] *GitHub - kensho-technologies/pyctcdecode: A fast and lightweight python-based CTC beam search decoder for speech recognition.* URL: https://github.com/kensho-technologies/pyctcdecode.

[45] Richard Liaw et al. „Tune: A Research Platform for Distributed Model Selection and Training“. In: *arXiv preprint arXiv:1807.05118* (2018).

[46] Tomasz Potapczyk und Paweł Przybysz. „SRPOL's System for the IWSLT 2020 End-to-End Speech Translation Task“. In: *IWSLT*. 2020.

[47] *bamerluk/Speaker_Diarization*. URL: https://github.zhaw.ch/bamerluk/Speaker%5C_Diarization.

Abbildungsverzeichnis

Tabellenverzeichnis

Anhang A

Code

A.1 Übersetzungsbeispiele

A.1.1 Szenario 1: Hybrid Dynamic Window

Ground-Truth:

Und er konnte sein Verlangen nicht mehr länger zurückhalten. Politisch interessieren mich Kultur und Jugendarbeit. Aber es ist eine gute Ausgangslage für den Demokraten. Er muss danach voraussichtlich rund zwei Monate pausieren. Eine Version für Konsolen und PC ist bereits in Arbeit. Namen will Jenny noch keine nennen. Auch eine Verdichtung an diesem Ort mit guter ÖV-Anbindung ist sinnvoll. Die Umsetzung des KiBeR werden wir sehr genau beobachten. Das dürfte eher ein lokales Phänomen sein. Die Gäste erlitten die erste Niederlage nach fünf Siegen. Weil es einen Teil ihres Lebens verändert. Das hat mich überhaupt nicht interessiert. Auch innerhalb der Lufthansa-Gruppe schaue man durchaus neidisch nach Zürich. Den habe ich zum Teil ausserordentlich stark erlebt. Die genaue Lage sei am Abend noch nicht abzuschätzen. Die Kraft wird jeweils per 6-Gang-Getriebe übertragen. Ohne das hätten wir das gar nicht hinbekommen. Denn damals wurden tatsächlich Neuheiten enthüllt. Ein Buch wird von Käufern wie ein Kunstwerk betrachtet. Einige Facebook-Freunde sind inzwischen echte Freunde geworden. Das neue Fahrzeug ist für praktisch sämtliche Aufgaben überdimensioniert. Die Hinweise seien von Anwohnern und über soziale Medien eingegangen. Dadurch würden die Schulden in den USA tendenziell weiter steigen. Er holte sich den Club ins Wohnzimmer. Kritikern zufolge wandten die Beamten zudem Gewalt gegen friedliche Demonstranten an.

Hypothese:

Und er hat sein Verlangen nicht mehr länger zurückhalten. Politisch interessieren mich Kultur und Jugendarbeit. Aber es ist eine gute Ausgangslage für den Demokrat. Er muss danach voraussichtlich rund zwei Monate pausieren. Eine Version für die Konsole und der PC sind bereits in Arbeit. Neman will Jenny noch keine nennen. Auch eine Verdichtung und Ort mit guter ÖV-A-Bindung ist sinnvoll. Die Umsetzung des KBR werden wir sehr genau beobachten. Das darf eher ein lokales Phänomen sein. Gestern haben die ersten Niederlage nach fünf Siegen erlitten. Weil ein Teil ihr Schlaben verändert. Das hat mich überhaupt nicht interessiert. Auch innerhalb der Lufthansa-Gruppe hat man durchaus auf Zürich. Da habe ich zum Teil ausserordentlich sparen lassen. Die genaue Age sei am Abend noch nicht abzusetzen. Die Kraft wird jeweils per <unk><unk>Gangbetrieb übertragen. Ohne die hätten wir das gar nicht hinbekommen. Die Damals wurden tatsächlich Neuheiten enthüllt. Ein Buch wird von den Käufern wie ein Kunstwerk betrachtet. Einige Facebook-Freunde sind mittlerweile echte Freunde geworden. Das neue Fahrzeug ist für praktisch sämtliche Aufgaben überdimensioniert. Die weise seien von Anwohnern und über soziale Medien eingegangen. Dadurch würden die Schulden in den USA tendenziell weiter steigen. Er hat sich der Club ins Wohnzimmer geholt. Kritiker zufolge hatten die Beamten zudem Gewalt gegen die friedlichen Demonstranten an.

A.1.2 Szenario 2: Hybrid Dynamic Window

Ground-Truth:

Das gigantische Unterfangen ist nicht ohne Kritiker. Heute sehe ich die Sache mit anderen Augen. Ich kann nicht einmal leicht joggen gehen. Das lassen sich die Briten allerdings auch ziemlich teuer bezahlen. Auch die kommende EM in Frankreich wird solche bieten. Hier geht es zum Voting. Der Umfang ist pro Jahr in der Grössenordnung von Fr. 30’000.00. Sie ist nachvollziehbar gestaltet und die Verbuchung der einzelnen Positionen ist korrekt. Bilder vom Shooting gibts nächste Woche in 20 Minuten.

Hypothese:

Das gigantische Unterfangen ist nicht ohne Kredit. Heute sehe ich die Dinge mit anderen Augen. Da an nicht einmal leicht zu schocken gehen. Das lassen sich die Briten

allerdings auch ziemlich teuer bezahlen. Auch die kommende EM in Frankreich will so etwas bieten. Da geht es um Voting. Der Umfang ist pro Jahr in der Grössenordnung von . ist nachvollziehbar gestaltet und die Verbuchung der einzelnen Positionen ist korrekt. Die Bildung des Shootings gibt es nächste Woche. Die in Minuten.

A.1.3 Szenario 3: Hybrid Dynamic Window

Ground-Truth:

Ich danke allen für die Mitarbeit. Und nicht jeder verläuft im gleichen Stil.
Es liegen zwei beantwortete Anfragen vor. Die Frau musste sich vor der Sendung einiges anhören.
Dazu duftet er wie eine Lilie. Heute lebt sie in Norddeutschland.
Und Nummer 6 ist mit der Weisheit des Alters überschrieben. Der Dollar seinerseits war an den Goldpreis fixiert.

Hypothese:

Ich danke allen für die Mitarbeit und nicht jeder Verlauf im gleichen Stil
Es liegen zwei beantwortete Anfragen vor Frau muss sich vor der Sendung einiges anhören
Dazu schmeckt er wie eine Linie, heute lebt sie in Norddeutschland.
Nummer ist mit der Weisheit des Alters überschrieben, der Dollar seinerseits war am Goldpreis fixiert.

A.1.4 Podclub: Hybrid Dynamic Window

Ground-Truth:

Liebe Zuhörerinnen und Zuhörer, Herzlich willkommen zur Sendung Zukker im Leben vom 3. November 2017. Es freut mich sehr, sind Sie wieder mit dabei. Brigit ist wieder zu Hause. Zusammen mit ihrem Freund Viktor habe ich ihr geholfen, die Wohnung aufzuräumen. Das war sehr eindrücklich, davon werde ich Ihnen heute gerne erzählen! Wandern Sie gerne, liebe Zuhörerinnen und Zuhörer? Ich wollte mich in der Natur entspannen, aber leider hatten viele Menschen die genau gleiche Idee wie ich

und es war alles andere als erholsam. Von meinem Ausflug auf die Rigi erzähle ich Ihnen ebenfalls heute! Viel Vergnügen! Ich habe Brigit nie im Krankenhaus besucht, weil ich eine neue Arbeit habe und leider keine Zeit hatte. Aber wir haben zweimal telefoniert und sie war ganz euphorisch am Telefon. Viktor besucht sie regelmässig. Viktor und Brigit kennen sich schon lange und Brigit mag ihn gut, aber irgendwann hat Viktor dann die beste Freundin von Brigit geheiratet. Dann hatten sie viele Jahre keinen Kontakt. Heute ist Viktor geschieden und Brigit und er schreiben sich Whatsapp-Nachrichten hin und her. Als sie ihm geschrieben hat, dass sie sich das Bein gebrochen hat, ist er sofort ins Krankenhaus gekommen. Brigit hat ihm erzählt, dass sie nichts wegwerfen kann, weil sie immer denkt, dass sie es noch brauchen kann. So wie die dreihundert Telefonbücher, oder die vielen Plastiksäcke. Sie schämt sich sehr, weil das nicht mehr ganz normal ist, weil das eine Krankheit ist. Zu Menschen, die zwanghaft Dinge sammeln sagt man Messi. Viktor hat ein Wunder vollbracht . Er hat ihr erklärt, dass es doch viel gemütlicher ist, wenn man nicht so viele Dinge besitzt, weil man sich dann immer wieder einmal etwas Neues kaufen kann. Und weil Brigit auch gerne Besuch hat, hat sie beschlossen , dass alles weggeworfen wird. Heute ist Brigit nach Hause gekommen und Viktor und ich stehen mit ihr in der Wohnung. Vor dem Haus steht eine Mulde , wo wir alles entsorgen . Brigit klammert sich an fünf Telefonbücher und sagt: Ihr dürft nicht alle wegwerfen! Viktor schaut sie an und sagt: Diese fünf kannst du behalten, aber die anderen werfen wir weg. Brigit steht auf ihrem Balkon, während Viktor und ich die Telefonbücher in die Mulde werfen. Dann sitzt sie in der Küche und hält sich die Augen zu, während ich die Plastiksäcke und die dreissig Salatschleudern mitnehme. Nach drei Stunden sieht die Wohnung viel besser aus. Brigit kommt mit einer Reisetasche aus dem Wohnzimmer und sagt: Die Stofftiere will ich nicht wegwerfen, die will ich verschenken. Ich sage zu ihr: Wenn wir das nicht heute machen, wirst du sie doch behalten. Wir könnten sie in ein Kinderheim bringen, was denkst du? Brigit findet es eine gute Idee. Viktor schaut mich an und sagt: Du musst nicht mehr helfen, du hast so viel gemacht, liebe Nora! Ich bin sehr froh, dass Viktor da ist und dass ich nicht mehr die einzige bin, die sich um Brigit kümmert. Ich verabschiede mich von den beiden und gehe in meine Wohnung. Die Mulde wird morgen abgeholt, ich hoffe, dass Brigit in der Nacht nicht heimlich wieder alle Sachen in die Wohnung holt. Ich schaue über das Geländer vom Balkon und sehe, wie Viktor Brigit an der Hand nimmt und sie die Reisetasche mit den Stofftieren trägt. Ich bin froh, dass Brigit einen Freund gefunden hat. Gehen Sie gerne wandern, liebe Zuhörerinnen und Zuhörer? Und wenn ja, haben Sie Tipps für mich, was ich unbedingt sehen sollte? Ich bin nämlich kein Wanderprofi. Ich lebe zwar schon immer in der Schweiz, aber ich kenne die Schweiz leider sehr schlecht. Meine beste Freundin hat mich überredet

und zusammen sind wir auf die Rigi gegangen. Die Rigi ist ein Berg in der Zentralschweiz und bei Touristen sehr beliebt. Darum sind wir extra am Freitag und nicht am Wochenende gegangen, weil es dann weniger Leute hat. Als wir in Art-Goldau die Zahnradbahn nehmen wollen, ist sie schon voll und wir müssen auf die nächste warten. Die Menschen vor uns drängeln und wir können nicht sitzen und müssen bis zur Bergstation stehen. Meine Freundin sagt: Wenn wir draussen sind, werden sich die Menschen verteilen und wir können gemütlich wandern. Ich bin genervt . Die Aussicht auf den Zugersee und den Vierwaldstättersee wäre schön, wenn nicht überall Menschen stehen würden, die alles fotografieren. Ich sehe nur durch die Bildschirme der iPhones auf die Seen. Ich sage zu meiner Freundin: Komm wir essen eine Bratwurst und trinken ein Bier. Sie findet das eine gute Idee. Im ersten Restaurant hat es keinen Platz. Im zweiten auch nicht. Nachdem wir eine halbe Stunde gelaufen sind, finden wir im dritten Restaurant einen Platz an der Sonne. Die Bratwürste sind leider schon kalt, als ich sie zu meiner Freundin an den Tisch bringe. Die Schlange vor dem Getränkestand war so lang, dass sie kalt wurden. Das ist alles nicht so, wie ich mir das vorgestellt habe. Vor der Luftseilbahn stehen sehr viele Menschen, die auch nach Weggis möchten. In der Gondel steht mir ein Mann auf den Fuss und seine Frau drückt mir ihren Rucksack ins Gesicht. Ich sage: Können Sie bitte aufpassen, ich habe keinen Platz! Die Frau sagt: Niemand hat hier Platz! Meine Freundin hat Angst, dass ich einen Streit anfange, aber dann sind wir zum Glück unten angekommen und gehen zum Steg , wo das Schiff nach Luzern fährt. Endlich! Genug Platz für alle. Wir sitzen draussen in der Sonne und schauen auf den Vierwaldstättersee. Meine Freundin lacht und sagt: Es tut mir leid, dass deine erste Wanderung so stressig ist! Ich lache sie an und sage: Ich will gar nicht wissen, wie viele Menschen es am Sonntag auf der Rigi hat! Wir lachen beide und sind dann sehr entspannt. Leider nicht lange. Als das Schiff in Luzern anlegt ist dort Jahrmarkt . Ich werde langsam aber sicher wirklich sauer und sage: Das darf doch nicht wahr sein, kann man denn nicht einfach seine Ruhe haben? Überall sind Kinder mit Ballonen, Zuckerwatte und überall ist laute Musik. Vor uns steht plötzlich ein Mädchen, das weint und sagt: Ich habe meinen Papa verloren! Wir nehmen sie an die Hand und bringen sie zum Häuschen vor dem Riesenrad, wo man Tickets kaufen kann. Der Mann im Häuschen ruft über den Lautsprecher den Papa des Mädchens aus. Wir warten mit ihr, bis der Papa kommt. Das Mädchen schaut meine Freundin an und fragt: Habt ihr beide schon lange nicht mehr geschlafen? Ihr seht sehr müde aus. Ich lache laut. Meine Freundin sagt: Es fährt in fünf Minuten ein Zug nach Zürich. Wir nehmen den Zug, fahren erschöpft nach Zürich und gehen etwas essen. Ausser uns hat es nur vier andere Menschen im Restaurant. Alle anderen sind sicher noch auf der Rigi. Ich freue mich sehr, wenn ich Ihnen am 17. November wieder auf podclub.ch und in der App aus meinem Leben erzählen darf. Dann ist die Sendung nur

auf Hochdeutsch. Ich bin auf eine Hochzeit eingeladen, davon werde ich ihnen sicher erzählen. Im Moment habe ich viel zu tun, aber ich habe Tom kurz in der Waschküche gesehen, über ihn erzähle ich Ihnen wieder. Schauen Sie doch in der Zwischenzeit bei Instagram unter PodClubNora und zukkerimleben vorbei und üben Sie mit dem Vokabeltrainer in unserer App. Und wenn Sie mir gute Tipps für schöne Wanderungen haben, schreiben Sie mir! Auf Wiederhören!

Hypothese:

Liebe Zuhörerinnen und Zuhörer Herzlich willkommen zur Sendung Zucker im Leben des d. November Es freut mich sehr sind sie wieder mit dabei. Brigit ist wieder die Hai. Zusammen mit ihrem Freund Victor habe ich ihn holen, die Wohnung aufzuräumen. Das war sehr eindrücklich . Davon werde ich Ihnen heute gerne erzählen. Wandern sie gerne bei Zuhörerinnen und Zuhörer Ich will mich in der Natur entspannen, aber leider haben viele Menschen die genau gleiche DK wie ich und es war alles andere als erholsam. Von meinem Ausflug auf Rigi erzähle ich Ihnen ebenfalls heute viel Vergnügen. Ich habe Brigit miei Krankenhaus besucht, weil ich eine neue Arbeit habe und leider keine Zeit hat. Aber wir haben zweimal telefoniert und sie war ganz euphorisch am Telefon. Der Victor besucht sie regelmässig. Viktor und Brigit kennen sich schon lange und Brigit mag ihm gut, aber irgendwann hat Victor dann die beste Freundin der Brigit geheiratet. Dann hatten es viele keinen Kontakt. Heute ist WiKDogeschiedenund bringt und schreiben sich und Azep-Nachrichten hin und her. Wo sie ihm geschrieben hat, dass sie sich bei gebrochen hat, ist er sofort ins Krankenhaus kommen. Brigit hat ihm erzählt, dass sie nicht wegwerfen kann weil sie immer an, dass sie es brauchen. So wie die Telefonbücher oder die vielen Plastik. Sie schämt sich und weil das nicht mehr ganz normal ist, weil das Krankheit ist. Zu Menschen, die zwanghaft Dinge sammeln, sagt man Messe. Victor hat das wundervoll gebracht. Man hat ihr erklärt, dass es doch viel gemütlicher ist, wenn man nicht so viele Dinge besitzt, weil man sich dann immer wieder einmal etwas Neues kaufen. weil Brigit auch gern besucht hat, hat sie beschlossen, dass alles weggeworfen wird. Heute ist SprigiTakm,undViktor und ich stimme mit ihr in der Wohnung. Vor dem Haus steht eine Mulde und bei allen sorgt. Brigid klammert sich an fünf Teile von Büchern und dürfen nicht alle wegwerfen. Victor schaut sie an und sagt die kannst du behalten, aber die anderen werden . Ich bringe die Stadt auf ihrem Balkon, während Victor und ich Telefonbüchern Mulden werfen. Dann sitzt sie in der Küche und hebt sich die Augen zu, während die Plastik und Salahledermitnehmn. Nach drei Stunden sieht die Wohnung viel besser aus. Die Frage Kunde mit einer Reisetasche aus dem Wohnzimmer

und Zeit. Die Stoffe will ich nicht wegwerfen will ich verschenken. Ich sage zu ihnen: Wenn wir das nicht heute machen, ist jedoch behalten. Wir könnten sie in ein Kinderheim bringen, was denkst du? finden Sie eine gute Idee. Victor schaut mich an und sagt, muss nicht mehr helfen, du hast so viel gemaht,liebeNora. Ich bin sehr froh, dass Victor ist und das ich nicht mehr die einzige bin, die sich um die Pregigt kümmern. verabschiede mich von den beiden und gehe in eine Wohnung. Die Mulde wird abgeholt, ich hoffe, dass die Brig alle Dinge in die Wohnung holt. Ich schaue über das Gelände des Balkons und sehe, wie Victor Brigit an der Hand nimmt und sie die Reisetasche mit dem Stofftier trägt. Ich bin froh, dass Brigit einen Freund gefunden hat. Können Sie wandern liebe Zuhörerinnen und Zuhörer. Und wenn ja haben sie Tipps für mich, was ich umbringen sehen. Ich bin nämlich kein Mann Profi. Ich lebe zwar schon immer in der Schweiz, aber ich kenne die Schweiz leider sehr schlecht. Meine beste Freundin hat mich überredet, und zusammen sind wir auf die Rigi gegangen. Riggi ist ein Berg in der Zentralschweiz und bei Touristen sehr belieben. Deshalb sind wir extra am Freitag und nicht am Wochenende weil es dann weniger Leute gab. Was wir in Gold und die SAradBahnbändenehmen ist sie schon voll und wir müssen auf die Nehstewalter. Die Menschen von uns drängen und wir können nicht Zeiten und müssen bis zu der Bergstation stehen. Meine Freundin sagt Wenn wir aussen sind, werden sich die Menschen verteilen und wir können gemütlich wandern. Ich bin genau. Die Aussicht auf den Zugersee und der Vierwaldstätter See wäre schön, wenn nicht überall Menschen stehen, die alles fotografiert. Ich sehe nur durch die Bilder von der Pons auf zusehen. Ich sage zu einer Freundin. Kaum wir essen der Braut wüst und trinken das Bier. Sie finden das eine gute Idee. Im ersten Restaurant hat es keinen Platz im zweiten auch nicht. Nachdem wir halbe Stunde gelaufen sind, finden wir im dritten Restaurant Platz an der Sonne. Bratwürste sind leider schon kalt, wo ich sie zu meiner Freundin am Tisch springen. Die Schlange vor dem Getränkestand war so lange, dass sie kalt . Das ist alles nicht so, wie ich mir das vorgestellt habe. Vor der Luftseilbahn stehen sehr viele Menschen, die auch noch wegen möchten. In der Gondel steht ein Mann auf den Fuss und seine Frau drückt ihre RücksackinsGesicht aufpassen keinen Platz. Frau sagt:Niemandhat Platz. Meine Freundin hat Angst, dass ich einen Streit ging, aber dann sind wir zum Glück akomundgezumSteg, da das Schiff nach Luzern fällt. Endlich Genug Platz für alle. Wir sitzen draussen in der Sonne und schauen auf den Vierwaldstädtesee. Meine Freundin lacht und zeigt es tut mir leid, dass seine erste Wanderung so stressig ist. Ich will gar nicht wissen, wie viele Menschen es am Sonntag auf der Rigi hat. Wir lachen beide und sind sehr entspannt. Leider nicht lange. Da das Schiff in Luzern anlegt, ist dort Kirby. Ich werde langsam aber sicher wirklich selber und sagen, dass doch nicht wohl sein kann man nicht einfach seine Ruhe haben. Überall sind Kinder mit Ballonen Zuckerwatte und überall ist laute Musik. Vor uns steht

plötzlich ein Mädchen, der und sagt: Ich habe mein Papa verloren. Wir nehmen sie an der Hand und bringen sie zum Häuschen vor dem Riesenrad, das an Tickets kaufen. Der Mann im Häusle ruft über die Lautsprecher oder Papa des Mädchen aus. Man wartet mit ihnen bis Papakun. Das Made meine Freundin und gefragt. Haben beide schon lange nicht mehr geschlafen sehen die Mut aus. Ich lache auf. Meine Freundin sagt, es fällt in Minuten ein Zug nach Zürich. Wir nehmen den Zug fahren die Schöpf nach Zürich und etwas gemessen. Ausser uns hat es nur vier andere Menschen im Restaurant. Alle anderen sind sicher noch auf der Regel. Ich freue mich sehr, wenn ich Ihnen am <unk>.Novemberwieder auf PutcluppunktChab und in den App aus meinem Leben zählen. Dann ist die Sendung nur auf Hochdeutsch. Ich bin auf eine Hochzeit eingeladen, davon werde ich Ihnen sicher erzählen. Im Moment habe ich viel zu tun, aber ich habe Tom kurz in der Woschocheeihnverzehe ich ihnen wieder. Schauen Sie doch in der Zwischenzeit bei Instagram. HertagPotClupNura und Herta Zucker im Leben vorbei und üben sie mit dem . Vocki-Trainer. In unser Und wenn Sie für mich gute Tipps für schöne Wanderungen haben schreiben Sie mich auf Widerlose.

A.2 Code Repository, Datensätze

Der komplette Code, die Installationsanleitung, sowie die erstellten Datensätze zur Evaluation, können unter folgendem Link gefunden werden:
`https://github.zhaw.ch/bamerluk/speech_translation`

A.2.1 README

Auf den nachfolgenden Seiten befindet sich das README, das auch im Code Repository zu finden ist.

README ST Pipeline, Datasets and Evaluations

Python Version 3.8 or 3.9 needed!

Linux Installation

you need git and git-lfs to install all necessary model etc.

```
sudo apt-get install git git-lfs

git clone ... <your-directory>
cd <your-directory>
git lfs pull
```

Optional Step: Create venv

```
# create venv
python3 -m venv <path-to-venv>

# activate venv before continuing with this README
source <path-to-venv>/bin/activate
```

Install dependencies

```
sudo apt-get update
sudo apt-get install build-essential gcc python3-dev

pip install wheel ffmpeg-python
pip install torch==1.11.0+cu113 torchvision==0.12.0+cu113
torchaudio==0.11.0+cu113 -f
https://download.pytorch.org/whl/cu113/torch_stable.html
pip install -r ./requirements.txt
```

Dataset Creation

If you use git-lfs, then the created datasets (scenario 1-3, SwissText and Podclub) will be downloaded in this repository (under ./data)

Create Datasets:

```
python3 ./preprocessing/synthetic/createSynthetic<DATASET>.py
```

To be able to create the datasets, you need to have the Base Datasets in the ./data directory (Base Datasets are not tracked in git)

- DialektSammlung
  - ./data/dialect
    - dialect.tsv
    - all clips (grouped by userid) need to be in the same directory level
- Podclub
  - ./data/podclub
    - /end_28/
      - all clips, that end with a 28 second outro
    - /end_50/
      - all clips, that end with a 50 second outro
- SNF
  - ./data/snf
    - testset
      - v0.2
        - export_v0.2.tsv
        - all clips (grouped by userid) need to be in the same directory level
- SwissText Testset
  - ./data/swiss_text
    - all.tsv
    - /clips/
      - all clips

Analysis

Plotting Windows

- Plotting the output of windows of the ST Pipeline can be done via helper functions in the file: **experiments/boundary.plot_beams.py**
- Examples of usages can be found in this folder: **experiments/boundary**

Token Analysis

- Various helpers to analyse token output can be found in the folders: **experiments/language_model** and **evaluation/plot**

Evaluation

Dataset Evaluation

All about dataset evaluation can be found in the **evaluation/dataset_evaluation** folder

Run Evaluation

- The different algorithms that translate audio to text can be found in this folder: **evaluation/**

- These algorithms are getting used by **./eval_model.py** and **./eval_calculated_timeslots.py**
  It can easily be specified for which datasets the evaluations should be run.
  The second file uses already saved timeslots for Speaker Diarization and SAD.
- For the dynamic and combined approaches those files can be found in:
  experiments/dynamic/eval_dynamic_model.py and
  experiments/combined/eval_combined_model.py
  For the moment, the combined approach only works with already saved timeslots.

Results Evaluation

All about results evaluation, creating plots etc, can be found in the **evaluation/results_evaluation** folder.

Hyperparameter

Run Hyperparameter search

```
# Hyperparameter search for Baseline
python -m experiments.hyperparameter_search_baseline

# Hyperparameter search for New Merging
python -m experiments.hyperparameter_search_baseline_new
```

View results from Hyperparameter search

To view and analyse Hyperparameter searches, you can use tensorboardX

```
pip install tensorboardX

# Baseline Parameter search
tensorboard --logdir=./experiments/hyperparameter_baseline/

# New Merging Parameter search
tensorboard --logdir=./experiments/hyperparameter_baseline_new/
```

A.3 Originale Aufgabenstellung

Zürcher Hochschule
für Angewandte Wissenschaften

School of
Engineering

Chuchichäschtli - Speech-to-Text für Schweizerdeutsch
BA22_ciel_05

BetreuerInnen:	Mark Cieliebak, ciel Jan Milan Deriu, deri
Fachgebiete:	Artificial Intelligence (AI) Machine Learning (ML)
Studiengang:	IT / WI
Zuordnung:	Centre for Artificial Intelligence (CAI)
Gruppengrösse:	2

Kurzbeschreibung:
In der Schweizer Dialektsammlung (www.dialektsammlung.ch) sammeln wir zurzeit Audio-Aufnahmen auf Schweizerdeutsch. Damit soll ein Speech-to-Text System (STT) entwickelt werden, das automatisch aus Schweizerdeutscher Sprache einen Text auf Hochdeutsch erzeugt.

Speech-to-Text Systeme funktionieren grundsätzlich gut und liefern auf Sprachen wie Englisch oder Hochdeutsch oft erstaunlich gute Transkripte. Wir haben bereits ein erstes System auf Schweizerdeutsch implementiert, das als Grundlage für diese Arbeit dient. Dieses System soll mit den neuen Daten aus der Dialektsammlung (mehrere hundert Stunden Audio in verschiedenen Dialekten) erweitert und optimiert werden.

Das Forschungsteam im Centre for Artificial Intelligence arbeitet zurzeit selbst intensiv an diesem Thema, unter anderem in einem Innosuisse-Projekt und einem Nationalfond-Projekt. Sie werden also an einem aktuellen Forschungsgebiet mitarbeiten und auch entsprechend viel mit unserem Team zusammenarbeiten.

Voraussetzungen:
* Die Implementierung ist in Python

* Die Lösung wird grosse Rechenleistungen zum Trainieren erfordern. Dafür kann unser GPU-Cluster verwendet werden

* Bereitschaft sich in ein herausforderndes Thema einzulesen und eine innovative Lösung zu implementieren

Falls Sie Interesse an diesem spannenden Thema haben, können wir gern einen Termin abmachen und die konkrete Aufgabenstellung besprechen. Email: ciel@zhaw.ch oder Telefon: 058 934 72 39.

Die Arbeit ist vereinbart mit:
Lukas Bamert (bamerluk)
Kevin Kläger (klaegkev)

Weiterführende Informationen:
http://Unter folgendem Link finden sie weitere Informationen zum Thema: https://interscriber.com/

Tuesday 07. June 2022 12:59