

Centre for Artificial Intelligence

Bachelor Thesis
Computer Science

Word Alignment for Swiss German ASR-Based Dialect Analysis: A Weighted Bipartite Matching Approach

Author	Micha Mettler
Main supervisor	Prof. Dr. Mark Cieliebak
External supervisor	Prof. Dr. Anja Hasse
Date	05.06.2026

Imprint

Project: Bachelor Thesis
Title: Word Alignment for Swiss German ASR-Based Dialect Analysis: A Weighted Bipartite Matching Approach
Author: Micha Mettler
Date: June 5, 2026
Keywords: swiss german, automatic speech recognition, word alignment, bipartite matching, dialect analysis
Copyright: Zurich University of Applied Sciences

Study program:
Computer Science, B.Sc.
Zurich University of Applied Sciences

Supervisor:
Prof. Dr. Mark Cieliebak
Zurich University of Applied Sciences
Email: ciel@zhaw.ch
Web: [Link](#)

External Supervisor:
Prof. Dr. Anja Hasse
University of Zurich
Email: anja.hasse@ds.uzh.ch
Web: [Link](#)

Abstract

Swiss German is a family of dialects that differs considerably from Standard German. The dialects also differ from one another in both pronunciation and vocabulary. Swiss German is, moreover, an almost exclusively spoken language, which makes text-based dialect analysis difficult. For this reason, this thesis presents an ASR-based approach to analyzing Swiss German dialects. Speech recordings, for which the read Standard German reference sentences are available, are transcribed by two different ASR models: one that understands Swiss German and one that was not trained on it. Where the two transcriptions diverge, the dialect-ignorant model fails to capture a dialectal word and instead produces a phonetic approximation, which serves as a signal of dialectal variation. To determine which words in the reference correspond to which words in the transcription, a sentence-level word alignment is required, which we formulate as a bipartite matching problem. We apply this alignment to the entire STT4SG-350 corpus and analyze the resulting word pairs from several angles. First, we measure the distance of each dialect region from Standard German and find that Wallis German is the most distant. Furthermore, we extract a dialect word lexicon, in which the vast majority of word pairs represent valid dialectal correspondences. Finally, through two case studies, we show the different variants that a single Standard German word takes across dialect regions. The results show that ASR-based word alignment provides a practical, extensible foundation for the data-driven study of Swiss German dialects.

Zusammenfassung

Schweizerdeutsch ist eine Gruppierung von Dialekten, die sich deutlich vom Hochdeutschen unterscheidet. Weiter gibt es innerhalb dieser Dialekte ebenfalls Unterschiede, sowohl in der Aussprache als auch im Wortschatz. Schweizerdeutsch ist ausserdem eine beinahe ausschliesslich gesprochene Sprache, was die Dialektanalyse auf Basis von Texten erschwert. Aus diesem Grund stellt diese Arbeit einen ASR-basierten Ansatz zur Analyse von Dialekten vor. Dabei werden Sprachaufnahmen, zu denen die vorgelesenen Referenzsätze auf Hochdeutsch vorliegen, von zwei unterschiedlichen ASR-Modellen transkribiert: einem, das Schweizerdeutsch versteht, und einem, das nicht auf Schweizerdeutsch trainiert wurde. Wo die beiden Transkriptionen voneinander abweichen, erfasst das dialektignorante Modell ein Dialektwort nicht und erzeugt stattdessen eine phonetische Annäherung, die als Signal für dialekt spezifische Wörter dient. Um die zusammengehörenden Wörter aus der Referenz sowie deren transkribierte Gegenstücke herauszufinden, braucht es ein satzweises Word-Alignment, welches in dieser Arbeit als bipartites Matching formuliert wird. Weiter wird dieses Word-Alignment auf den gesamten STT4SG-350-Korpus angewandt, um auf Basis der Ergebnisse unterschiedliche Dialektanalysen durchzuführen. Zunächst werden die Distanzen jeder Dialektregion zum Hochdeutschen gemessen, wobei Walliserdeutsch, der Dialekt im Kanton Wallis, die grösste Distanz aufweist. Ausserdem wird auf Basis des Word-Alignments ein Dialektwörterlexikon erstellt, bei welchem ein Grossteil der Wortpaare valide Dialektwörter repräsentiert. Abschliessend werden zudem zwei Fallstudien durchgeführt, welche die verschiedenen phonetischen Annäherungen der Dialektregionen für ein Referenzwort aufzeigen. Insgesamt zeigen die Resultate, dass ein ASR-basiertes Word-Alignment eine anwendbare und erweiterbare Grundlage für datenbasierte Dialektanalyse darstellt.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Prof. Dr. Mark Cieliebak, for the continuous guidance, support, and engagement throughout this thesis. Our regular discussions and his thoughtful feedback were invaluable in shaping both the direction and the quality of this work. I am also grateful to Andreas Mohn for the insightful discussions and for suggesting the analytical approach. Finally, I thank Damaris Mettler and Nadir Ferlin for proofreading this work.

Use of Generative AI

Generative AI systems and AI tools were used in various phases of this thesis. Specifically, Claude was used to brainstorm, proofread the text, format LaTeX content such as formulas and tables, and help write program code. In particular, the idea of encoding each alignment as a concatenated reference–hypothesis token for the TF-IDF analysis ([subsection 8.2.1](#)) was suggested by Claude. GitHub Copilot was used for merge reviews.

Contents

Abstract	iii
Zusammenfassung	iv
Acknowledgements	v
Use of Generative AI	vi
List of Figures	ix
List of Tables	x
List of Abbreviations	xi
List of Symbols	xii
1 Introduction	1
2 Theoretical Foundations	2
3 Related Work	4
4 Approach	6
4.1 DIT and DAT	6
4.2 Dataset	8
4.2.1 Dialect Regions	8
4.3 ASR Models	9
4.3.1 WER	10
5 Word Similarity	11
5.1 Lexical Similarity	11
5.2 Positional Similarity	12
5.3 Combined Similarity	12
5.4 Cross-Comparison	13
6 Word Alignment	17
6.1 Weighted Bipartite Matching	17
6.1.1 Partitions	17
6.1.2 Formal Definition	19
6.1.3 Cost Functions	19
6.1.4 Optimization	20
6.1.5 Implementation	21
6.2 Hyperparameters	22

6.2.1	Test Samples	22
6.2.2	Ground Truth	22
6.2.3	Grid Search	24
6.2.4	Analytical Approach	27
7	Aligning the Dataset	29
7.1	Data	29
7.2	Preprocessing	29
7.3	Alignment	30
7.3.1	Implementation	31
8	Dialect Analysis	32
8.1	Regional Distance	32
8.1.1	Measuring Distance to Standard German	32
8.1.2	Wallis German: The Most Distant, and Highest, Alemannic	33
8.2	Dialect Word Lexicon	35
8.2.1	Identifying Region-Specific Word Pairs	35
8.2.2	Lexicon	35
8.2.3	DAT as Reference	39
8.2.4	Implementation	39
8.3	Case Studies	39
8.3.1	Anke	39
8.3.2	Geweint	41
8.4	Interactive Tool	43
8.4.1	Regional Distance Page	43
8.4.2	Dialect Word Lexicon Page	43
8.4.3	Reference Word Search	44
8.4.4	Reference Word Detail View	45
9	Discussion and Future Work	46
9.1	Discussion	46
9.2	Future Work	47
10	Conclusion	48
A	Appendix A: Detailed Results	49
A.1	Ground-Truth Sample Set	49
A.2	Dialect Word Lexicon	49
A.2.1	Reference vs. DIT	50
A.2.2	DAT vs. DIT	52
B	Appendix B: Code Listings and Manuals	54
	Bibliography	57
	Glossary	60

List of Figures

4.1	Pipeline overview	7
4.2	Dialect regions of the STT4SG-350	8
5.1	Distribution of sim_{lex} , sim_{pos}	15
6.1	Bipartite graph for example (1)	17
6.2	Augmented bipartite graph for the insertion example in (2)	18
6.3	Full min-cost max-flow network for the insertion example (2)	19
6.4	Optimal flow assignment for the insertion example (2)	21
6.5	Screenshot: Terminal Tool for Ground Truth Annotation	23
8.1	Mean total cost per dialect region	33
8.2	Regional dialect distance	33
8.3	DIT variants of <i>Butter</i> across regions	39
8.4	Bipartite flow network for the <i>Anke</i> example	41
8.5	DIT variants of <i>geweint</i> across regions	41
8.6	Regional Distance page of the interactive tool	43
8.7	Dialect Word Lexicon page of the interactive tool	44
8.8	Search view of the Dialect Word Lexicon page	44
8.9	Detail view of the Dialect Word Lexicon page	45

List of Tables

4.1	STT4SG-350 corpus statistics per dialect region	9
4.2	STT4SG-350 corpus statistics per split	9
4.3	Overall WER	10
4.4	WER per dialect region	10
5.1	Cross-comparison for reference word <i>nun</i>	14
5.2	Mean sentence length (in words) across 100 Wallis samples	14
6.1	Full Grid Search Results	25
6.2	Per-section optima at fixed configuration modes	26
6.3	Selected hyperparameters for the final alignment model.	26
6.4	Per-section accuracy at selected hyperparameters	26
7.1	Preprocessing normalizations	30
7.2	Stored alignment format	30
8.1	Regional distance from Standard German	34
8.2	Regional distance including preterite sentences	34
8.3	Selected dialect word pairs	36
8.4	Invalid dialect word pairs	38
8.5	DIT variants of <i>Butter</i> : counts and scores	40
8.6	DIT variants of <i>geweint</i> : counts and scores	42
A.1	Clip IDs of the ground-truth sample set	49
A.2	Dialect word pairs (TF-IDF)	50
A.3	Dialect word pairs (DAT-DIT comparison)	52
B.1	Python Dependencies	56

List of Abbreviations

ASR	Automatic Speech Recognition
DAT	Dialect-Aware Transcript
DIT	Dialect-Ignorant Transcript
TF-IDF	Term Frequency–Inverse Document Frequency
WER	Word Error Rate

List of Symbols

r, h	Reference and hypothesis sentence	
r_i, h_j	i -th reference word, j -th hypothesis word	
$ r , h $	Length of reference/hypothesis sentence (in words)	
$ r_i , h_j $	Length of word r_i / h_j (in characters)	
d_L	Levenshtein distance between two words	
$L_{\max}^{\text{word}}$	Maximum word length in the dataset	
sim_{lex}	Lexical similarity	$[0, 1]$
sim_{pos}	Positional similarity	$[0, 1]$
sim_{comb}	Combined similarity	$[0, 1]$
R, H	Set of reference/hypothesis words	
R', H'	Augmented partitions including ε -nodes	
n_r, n_h	Number of reference/hypothesis words ($= r , h $)	
N	Augmented partition size, $N = R' = H' $	
G	Bipartite flow network	
V, E	Vertex and edge set of G	
s, t	Source and sink nodes of the flow network	
M_{solver}	Reduced matching produced by the bipartite solver	
M_{gt}	Ground-truth matching (manually annotated)	
c_{ij}	Cost of edge from node i to node j	
f_{ij}	Flow on edge from node i to node j	$\{0, 1\}$
C	Total cost of a flow assignment	
T	Set of reference/hypothesis pairs in evaluation subset	
F	Analytical proxy function for hyperparameter optimization	
α	Lexical-positional weighting hyperparameter	$[0, 1]$
λ	Insertion/deletion penalty hyperparameter	$[0, 1]$
α^*	Optimal α from grid search or analytical solution	
λ^*	Optimal λ from grid search or analytical solution	
ε	Auxiliary node absorbing unmatched words	

Chapter 1

Introduction

Swiss German is a family of dialects spoken across the German-speaking regions of Switzerland. It differs from Standard German in phonology, vocabulary, morphology, and syntax. As it is primarily a spoken language, it is also a low-resource language for speech technology, which makes the systematic study of how its dialects differ a genuine challenge [1].

In recent years, the creation of large speech corpora has made data-driven dialect analysis approaches to Swiss German increasingly feasible. These corpora pair spoken Swiss German audio recordings with written reference texts, opening the door to studying dialectal variation at scale. At the same time, Automatic Speech Recognition (ASR) systems have improved substantially [2], [3].

This thesis combines these two developments. We propose an ASR-based approach to analyzing Swiss German dialects, built on the following idea: if a recording is transcribed once by a dialect-aware model and once by a model that does not understand the dialect, the difference between the two transcriptions reveals where the dialect diverges from Standard German. Concretely, where a dialectal word differs from its Standard German form, the dialect-ignorant model cannot transcribe it correctly and instead produces a phonetic approximation. These approximations are the signal we analyze.

To make use of this signal, we need to know which word in a transcription corresponds to which word in the reference. This is a word alignment problem, and solving it is the methodological core of this thesis. We formulate the alignment as a weighted bipartite matching between the words of two sentences. Computing this alignment requires a measure of how similar two words are, which we introduce by combining lexical and positional similarity. The resulting pipeline aligns the transcriptions of an entire corpus, producing a dataset of word pairs for analyzing dialectal variation.

Building on this dataset, we conduct a dialect analysis in three parts. We examine the distance of each dialect region to Standard German, present a dialect word lexicon derived from our data, and pursue two case studies. For the analysis, we provide an interactive, openly accessible web tool.

Chapter 2

Theoretical Foundations

Strictly speaking, Swiss German is not a single language variety of its own but a bundle of dialects spoken within Switzerland. A variety is a distinct, clearly separable form of a language, separated, for example, regionally, socially, or situationally. Swiss German, by contrast, is not uniform: it denotes the Alemannic dialects spoken in the southwestern part of the German-speaking area. Standard German, on the other hand, is the standardized variety used in formal writing and speech across the German-speaking countries [1].

Within Alemannic, the dialects form a continuum ranging from Low Alemannic in the north (such as the traditional dialect of Basel), through High Alemannic (Zurich, Ostschweiz), to Highest Alemannic in the southern Alpine regions, of which Wallis German, the dialect spoken in the canton of Wallis, is the most prominent example [1]. The distinction between High Alemannic and Highest Alemannic is characterized by relics of earlier linguistic features; for instance, the *hiatus*, in which two vowels meet without a consonant between them and are pronounced separately. The Highest Alemannic dialects preserve the old sounds such as *schniie*, *buue*, while the High Alemannic regions pronounce them with a diphthong (a vowel gliding from one sound to another) as *schneie*, *bouue* [1].

Furthermore, differences between the dialects are often characterized by west–east and north–south contrasts. The north–south contrast is linguistically associated with the geographic isolation of the southern Alpine regions: modern linguistic innovations in the *Mittelland* (the Swiss lowland plateau) reached the remote mountain regions only with difficulty [1].

One important characteristic shared across the Swiss German dialects is the absence of the preterite tense (*Präteritumschwund*). The simple past of Standard German does not exist in Swiss German; the perfect tense is used in its place [4], [5], [6]. This is illustrated in example (1).

- | | | |
|-----|-------------------------|------------------------------|
| (1) | <i>Es schneite.</i> | (reference, preterite) |
| | <i>Es het gschneit.</i> | (spoken in dialect, perfect) |

Beyond phonology and grammar, Swiss German also differs lexically. Many words are realized as entirely different forms than in Standard German, such as the dialectal *Anke* for *Butter* or *Geiss* for *Ziege* [1]. Furthermore, as French is another national language of Switzerland, Swiss German contains loanwords such as *Trottoir* (from

French) for Standard German *Bürgersteig* [7]. These lexical differences are the focus of our dialect word analysis in [chapter 8](#).

Furthermore, Swiss German is primarily a spoken language with no standardized orthography. As a low-resource language, it has historically lacked the large annotated datasets needed for speech technology [7].

Chapter 3

Related Work

The foundation for any data-driven study of Swiss German is the availability of annotated speech data. Recent corpora include the SDS-200 [2], around 200 hours of Swiss German speech, and the STT4SG-350 [3], 343 hours balanced across all seven dialect regions defined by the corpus. The balanced design of the STT4SG-350 makes it particularly suited for comparing dialects, which is why we base our analysis on it [2], [3].

In dialectometry, the quantitative study of dialect variation, Scherrer and Stoeckle [8] apply dialectometric methods to Swiss German, comparing dialects across phonological, lexical, morphological, and syntactic levels using data from the Swiss German dialect atlases. They confirm findings from traditional dialectology [9] — such as the two major geographical west–east and north–south divisions, as well as the more “conservative” southern dialects, especially in the Wallis — while also allowing for a more precise characterization of the dialectal landscape [8].

The work of Paonessa et al. [7] investigates dialect transfer for Swiss German speech translation on the same STT4SG-350 corpus that we use. Using dialectometric distances derived from Swiss German dialect-atlas data, as presented by Scherrer and Stoeckle [8], they find Wallis to be the most distant dialect and confirm this empirically through transfer experiments: leaving Wallis out of training causes the largest performance drop across all their models, and training on Wallis alone generalizes the worst to the other regions [7].

This thesis builds most directly on two ZHAW (Zurich University of Applied Sciences) student projects on the same problem. Berthele and Spangenberg [10] analyze Swiss German dialects by comparing ASR transcriptions against a Standard German reference, introducing a word-by-word *Retrieval Rate* (RR) based on the Levenshtein distance [11] and using it both for region-level comparisons and for identifying dialect-specific words. Like the studies above, they find Wallis to be the clear outlier and document the Whisper large-v2 translation behavior discussed in [section 4.1](#). We reuse several of their contributions directly: their comparison of ASR models, on which we base our choice of Whisper large-v2; their transcriptions; and their manual annotations marking the preterite case and the usability of individual clips. For identifying dialect-specific words, however, they report that thresholding the retrieval rate yielded unsatisfactory results, as a hard cutoff sits poorly on the continuum from dialect to accent to standard language. Instead of thresholding a single score, we introduce a word alignment method that determines which reference word corresponds to which hypothesis word, thereby enabling the positional

analysis of word changes across dialects, which they identified as an open direction but could not pursue [10].

Brell [12] approaches the same problem with an optimized word-pair matching method based on similarity scores, reporting that individual matches surface meaningful dialect correspondences, but that fully automated identification of dialect-specific words remains difficult, with a reported precision of around 26% on a restrictive set. Our method differs by using an alignment based on weighted bipartite matching, solved as a minimum-cost flow problem, and by applying it to the full `train_all` split rather than a restrictive subset [12].

Chapter 4

Approach

In this thesis, we examine a large corpus of spoken Swiss German across multiple dialect regions (described in [section 4.2](#)). To analyze these audio recordings, we choose an approach based on Automatic Speech Recognition (ASR).

4.1 DIT and DAT

Concretely, we analyze Swiss German based on the outputs of ASR transcriptions. For this, we use two different ASR models: one considered dialect-ignorant, producing a **dialect-ignorant transcript (DIT)**, and one that is dialect-aware, producing a **dialect-aware transcript (DAT)**. The DIT model is not trained on Swiss German and therefore should not understand the dialect. The DAT model, on the other hand, is trained on Swiss German and should therefore transcribe the dialect correctly, acting as our baseline. The transcribed output of an ASR model is referred to as the **hypothesis**.

During the transcription process, the DIT model is expected to understand the parts of the spoken sentence that are close to Standard German and to transcribe them correctly, producing a hypothesis that matches the reference. However, the parts it does not understand are the **dialect-specific words**, which differ from Standard German in the regional dialect. For these, the DIT model produces phonetic approximations of the spoken form.

For instance, the reference sentence

(1) *Anne isst Brot.*

contains no dialect-specific words and is expected to be transcribed correctly. With a dialect word, the picture changes:

(2)	<i>Anne isst Brot mit Butter.</i>	(reference)
	<i>Anne isst Brot mit Anke.</i>	(spoken in dialect)

In Swiss German dialect, the word *Butter* changes to the dialectal form *Anke*. The DIT's phonetic approximation is now a word that sounds similar to *Anke* but is not *Anke* itself, since the DIT model does not recognize it. An example DIT hypothesis looks as follows:

(3) *Anne isst Brot mit Anker.*

(DIT hypothesis)

The form *Anker* is the DIT’s phonetic approximation of the dialectal *Anke*, which we refer to as the **DIT variant**.

These DIT variants are the basis for our dialect analysis. However, to determine which reference word corresponds to which hypothesis word, we need to **align** the two. For this, we implemented a sentence-level word alignment pipeline, which forms the methodological core of this thesis. Concretely, we use word similarities and bipartite matching [13]; the full method is described in [chapter 5](#) and [chapter 6](#).

For an outline, the full pipeline is summarized in [Figure 4.1](#). The preprocessing and corpus-wide alignment are described in detail in [chapter 7](#).

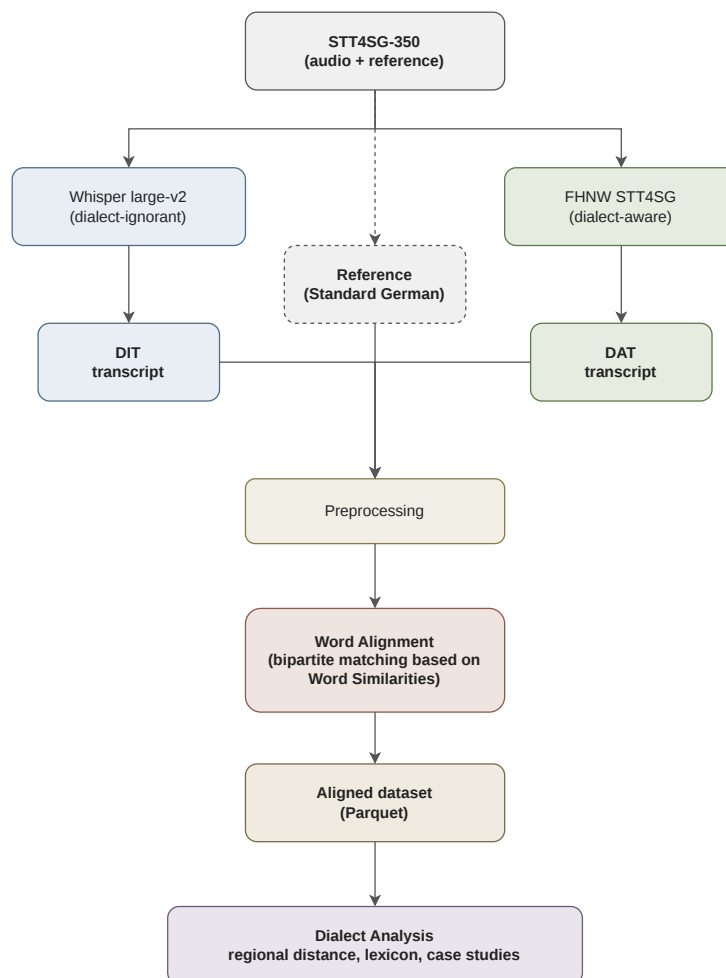


FIGURE 4.1: Overview of the ASR-based dialect analysis pipeline. The audio recordings of the STT4SG-350 dataset are transcribed by two ASR systems, producing the DIT and DAT transcripts. Together with the Standard German reference, these are preprocessed and aligned using bipartite matching, yielding an aligned dataset on which the dialect analysis is conducted. The individual stages are described in the following chapters.

4.2 Dataset

This thesis is based on two datasets, the STT4SG-350 [3] and the SDS-200 [2]. Both contain large corpora of Swiss German speech, annotated with Standard German text at the sentence level. This annotated Standard German text is referred to as the **reference**: the text that the speakers were given to read aloud in their Swiss German dialect. The STT4SG-350 contains roughly 343 hours of speech from 316 speakers, while the SDS-200 consists of 200 hours of speech from around 4,000 speakers [2], [3].

The STT4SG-350 provides a balanced amount of speech per dialect region. For this reason, it serves as the basis for our analysis. Specifically, its test set consists of the same spoken sentences for each dialect region, which was particularly useful for the hyperparameter selection described in section 6.2 [3].

4.2.1 Dialect Regions

As stated in chapter 2, there are no clear borders between dialect regions. Nonetheless, to make the data categorizable, the authors of the STT4SG-350 corpus divided Swiss German into seven dialect regions based on the clustering method of Scherrer and Stoeckle [8]. As we base our analysis on this dataset, we use the same dialect regions. These are visualized in Figure 4.2.

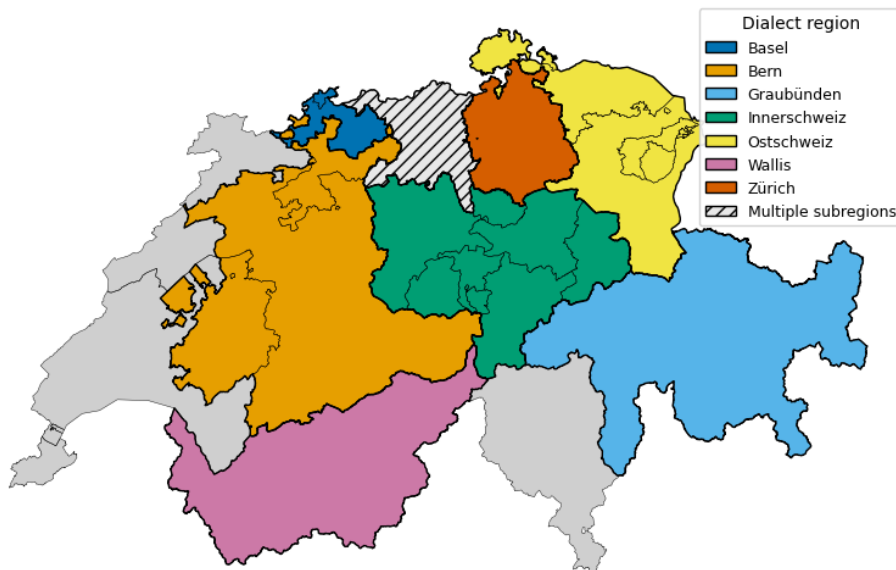


FIGURE 4.2: Approximate cantonal extent of the seven dialect regions of the STT4SG-350, following the canton correspondence given by Paonessa et al. [7]. Cantonal boundaries in this figure are based on the swissBOUNDARIES3D dataset by the Federal Office of Topography [14]. Aargau, which the corpus splits across several regions, is shown with a striped pattern. As the corpus defines the regions via dialectometric clustering with intentionally fuzzy borders [3], the cantonal rendering shown here is an approximation.

However, our region names follow the German naming used throughout this thesis. The dataset itself uses the English names Basel, Bern, Grisons (Graubünden), Central

(Innerschweiz), Eastern (Ostschweiz), Valais (Wallis), and Zurich [3]. The dialect regions are also reported in Table 4.1.

TABLE 4.1: Corpus statistics per dialect region, as reported by the STT4SG-350 dataset [3]. The population figure is an approximation as of 2023 and includes only German-speaking inhabitants.

Region	Population	Hours	Recordings	Speakers
Basel	0.4 M	47.5	34,169	44
Bern	1.2 M	48.7	35,683	46
Graubünden	0.2 M	44.3	30,931	46
Innerschweiz	0.8 M	49.1	36,402	43
Ostschweiz	0.9 M	52.6	38,182	47
Wallis	0.1 M	51.8	36,457	44
Zurich	1.6 M	49.3	35,703	46

Furthermore, the corpus statistics per split are displayed in Table 4.2, as reported by Plüss et al. [3].

TABLE 4.2: Corpus statistics per split, as reported by the STT4SG-350 dataset [3]. For the training set, the balanced version is given in parentheses.

	train_all (bal.)	valid	test	full
Hours	276 (239)	34	34	343
Recordings	200K (173K)	23K	25K	248K
Unique sentences	192K (167K)	23K	4K	218K
Speakers	219 (192)	21	76	316
Avg. rec./speaker	912 (902)	1106	324	783

4.3 ASR Models

As the next step, the ASR models for the DIT and DAT are selected. The selection of the DIT turned out to be a challenge since models that reliably capture the DIT variants tend to also struggle with the rest of the sentence, which must be preserved. We carried out some exploratory comparisons of the different Whisper models but then decided to rely on the extensive analysis already done by Berthele and Spangenberg [10]. They chose Whisper large-v2 [15], which overall produced the best results in their analysis. In addition, we were able to reuse their transcriptions along with their annotations, such as marking which sentences are in the preterite and whether the audio recordings are usable [10]. The use of these annotations is further explained in chapter 7.

Whisper large-v2 is a multilingual model trained on many different languages, none of which is Swiss German [15]. Still, Berthele and Spangenberg note an important limitation that surfaced during their analysis: the model occasionally translates words instead of merely transcribing them. As a result, some dialect-specific words are lost, as the model transcribes them back to their Standard German form instead of producing a phonetic approximation. Given the reliable transcription of the rest of the sentence, the model still suits our approach. Nonetheless, this limitation should be kept in mind [10].

For our DAT baseline, we chose the FHNW Swiss German Whisper model [16], a Whisper model fine-tuned on Swiss German data, including the STT4SG-350 corpus [3]. Using the model’s API [17], we transcribed both the STT4SG-350 corpus and the SDS-200. The resulting transcripts are available in the thesis’s GitHub repository, linked in [Appendix B](#).

4.3.1 WER

To quantify how well each transcript matches the Standard German reference, we use the **Word Error Rate (WER)** [18], a standard metric for evaluating ASR systems. The WER is based on the Levenshtein distance [11] at the word level: it counts the substitutions S , deletions D , and insertions I needed to transform the hypothesis into the reference, normalized by the number of words N in the reference [18]:

$$\text{WER} = \frac{S + D + I}{N}. \quad (4.1)$$

We computed the WER of both the DAT and the DIT against the Standard German reference using the `jiwer` library [19]. The evaluation was restricted to clips marked as usable by Berthele and Spangenberg [10]. In addition, 1,249 clips that produced errors during the DAT transcription process were excluded from both WER computations. This leaves 198,274 clips across all seven dialect regions. The overall results are reported in [Table 4.3](#). Additionally, the results for each dialect region are reported in [Table 4.4](#).

TABLE 4.3: Overall Word Error Rate of the DAT and DIT against the Standard German reference, computed over 198,274 evaluated clips across all seven dialect regions.

Transcript	WER
DAT (FHNW)	0.156
DIT (Whisper large-v2)	0.269
Δ (DIT – DAT)	0.113

TABLE 4.4: Word Error Rate of the DAT and DIT per dialect region, with their difference Δ . Here, n denotes the number of evaluated clips per region. All values are rounded to three decimal places; Δ is computed from the unrounded WERs and may therefore differ from the difference of the rounded columns by up to 0.001.

Region	n	DAT WER	DIT WER	Δ (DIT – DAT)
Basel	27,177	0.130	0.245	0.115
Bern	28,514	0.139	0.252	0.113
Graubünden	23,939	0.146	0.263	0.116
Innerschweiz	29,352	0.130	0.226	0.096
Ostschweiz	31,155	0.144	0.276	0.131
Wallis	29,426	0.165	0.375	0.209
Zürich	28,711	0.136	0.239	0.103

As expected, the DAT achieves a lower WER than the DIT across all regions. These results confirm that the two selected ASR models suit our approach.

Chapter 5

Word Similarity

To determine which reference word r_i matches which hypothesis word h_j , we need a metric describing how close the two words are. For this, we use two sub-metrics: the lexical similarity sim_{lex} (Equation 5.1) and the positional similarity sim_{pos} (Equation 5.2), which are then combined into a weighted similarity sim_{comb} (Equation 5.3).

The lexical similarity is the intuitive choice, matching words with similar character sequences. However, there are two reasons why the lexical similarity is not enough and the position inside the sentence is crucial for our matching: First, Swiss German offers no guarantee that words are written similarly to Standard German; dialect-specific words, which are ultimately what we are most interested in, can differ completely from their Standard German counterparts. Second, if the same hypothesis word appears more than once in a sentence, lexical similarity alone cannot distinguish between the occurrences.

5.1 Lexical Similarity

As the first sub-metric, we introduce our lexical similarity sim_{lex} . This sub-metric is based on the Levenshtein distance [11], i.e., the minimum number of character-level insertions, deletions, and substitutions required to transform one word into another.

However, the raw Levenshtein distance d_L [11] does not account for the length of the words. For example, a single-character difference in a long word like Fussballstadion $\rightarrow$ Fussballstation produces the same $d_L = 1$ as Haus $\rightarrow$ Maus, although the substitution has a greater impact on the short word. We therefore normalize d_L , either globally based on the maximum word length in the dataset $L_{\text{max}}^{\text{word}}$ or locally based on the combined length of the two words being compared, $|r_i| + |h_j|$, where $|r_i|$ and $|h_j|$ denote the number of characters in the reference and hypothesis word. The motivation for global normalization is to make sim_{lex} comparable across different pairs of words and across sentences, which is especially relevant in the cross-comparison in section 5.4. However, as a downside, dividing by the rather large $L_{\text{max}}^{\text{word}}$ compresses the range of values, leaving most similarities clustered near 1. Which setting performs better is evaluated in the grid search in section 6.2.3.

Finally, we subtract the normalized distance from 1 to invert the scale: high values of sim_{lex} now correspond to low Levenshtein distances and, thus, to words that are close to each other.

$$\text{sim}_{\text{lex}}(r_i, h_j) = \begin{cases} 1 - \frac{d_L(r_i, h_j)}{L_{\text{max}}^{\text{word}}} & \text{if global normalization} \\ 1 - \frac{d_L(r_i, h_j)}{|r_i| + |h_j|} & \text{if local normalization} \end{cases} \quad (5.1)$$

By construction, sim_{lex} always lies in $[0, 1]$, which makes it comparable to sim_{pos} .

5.2 Positional Similarity

For the positional similarity sim_{pos} , we want to find out how close the reference word is to the hypothesis word in terms of sentence position. To achieve this, we simply count the gap between the word positions. We include this sub-metric based on the assumption that a dialectal variation appears at a similar position in the hypothesis h as its counterpart in the reference r . Sentence position, therefore, serves as an additional signal for determining how similar two words are. We evaluate how well this assumption performs in the dialect word analysis (section 8.2).

Concretely, the gap is computed as the absolute difference between the two indices, normalized by $\max(|r|, |h|) - 1$, which is the largest possible gap given the number of words in the reference sentence ($|r|$) and the hypothesis sentence ($|h|$). Taking the maximum ensures that the denominator is always a valid upper bound on the positional gap, even when the two sentences differ in length. For instance, in two sentences of 5 words each, the word at index 0 and the word at index 4 produce a gap of 4, which normalizes to 1.

Similar to sim_{lex} , there are two settings for computing the positional similarity. The similarity is either returned directly (linear) or squared (Equation 5.2). With the squared setting activated, small gaps remain close to 1, while medium and large gaps are penalized more. Which setting performs better is evaluated in the grid search (section 6.2.3). Finally, we subtract the normalized gap from 1 to invert the scale, analogous to sim_{lex} . Through this, words at the same position produce a sim_{pos} of 1, and the two sub-metrics sim_{lex} and sim_{pos} become comparable.

$$\text{sim}_{\text{pos}}(r_i, h_j) = \begin{cases} 1 & \text{if } \max(|r|, |h|) = 1 \\ 1 - \frac{|i - j|}{\max(|r|, |h|) - 1} & \text{if linear} \\ \left(1 - \frac{|i - j|}{\max(|r|, |h|) - 1}\right)^2 & \text{if squared} \end{cases} \quad (5.2)$$

Again, sim_{pos} always lies in $[0, 1]$.

5.3 Combined Similarity

After defining the two sub-metrics sim_{lex} and sim_{pos} , we want to combine them into a single similarity score. We do this via a weighted sum controlled by the parameter α . A high α favors sim_{lex} ; a low one favors sim_{pos} . The formula can also be reduced to one side via the boundary values: setting $\alpha = 1$ leaves sim_{lex} alone, while $\alpha = 0$

leaves sim_{pos} alone. The idea behind $\text{sim}_{\text{comb}}(r_i, h_j)$ is based on the weighted combination of scores described by Manning et al. [20].

$$\text{sim}_{\text{comb}}(r_i, h_j) = \alpha \cdot \text{sim}_{\text{lex}}(r_i, h_j) + (1 - \alpha) \cdot \text{sim}_{\text{pos}}(r_i, h_j), \quad \alpha \in [0, 1] \quad (5.3)$$

5.4 Cross-Comparison

For an initial exploratory analysis, we conducted a cross-comparison on a small subset of the data to inspect how our similarity measures behave. We used 100 samples from the Wallis dialect region of the STT4SG-350 dataset [3], the same subset used in subsection 6.2.1, whose clip IDs are listed in Table A.1.

Procedure The procedure, summarized in Algorithm 1, works as follows: every reference word r_i is compared against every hypothesis word h_j , leading to an exhaustive comparison of all possible word combinations. For each pair, we compute both the lexical and positional similarities, as well as their combined similarity sim_{comb} . Alongside the similarities, we also store the word forms and their sentence positions, which enables both quantitative analysis through distribution plots (Figure 5.1) and qualitative inspection of individual pairs.

Algorithm 1 Cross-Comparison

Require: Reference words r , hypothesis words h , weighting parameter α

Ensure: Set of similarity records S

```

1:  $S \leftarrow \emptyset$ 
2: for  $i \leftarrow 1$  to  $|r|$  do
3:   for  $j \leftarrow 1$  to  $|h|$  do
4:      $s_{\text{lex}} \leftarrow \text{sim}_{\text{lex}}(r_i, h_j)$ 
5:      $s_{\text{pos}} \leftarrow \text{sim}_{\text{pos}}(r_i, h_j)$ 
6:      $s_{\text{comb}} \leftarrow \alpha \cdot s_{\text{lex}} + (1 - \alpha) \cdot s_{\text{pos}}$ 
7:      $S \leftarrow S \cup \{(i, j, r_i, h_j, s_{\text{lex}}, s_{\text{pos}}, s_{\text{comb}})\}$ 
8:   end for
9: end for
10: return  $S$ 

```

For the cross-comparison, we used the initial hyperparameters $\alpha = 0.7$, weighting lexical similarity higher and global lexical normalization with $L_{\text{max}}^{\text{word}} = 26$ (the longest word length in our 100 samples), since we wanted sim_{lex} to be comparable across word pairs and across sentences. Furthermore, the positional similarity was returned directly, without the squared setting activated. This suited our exploratory analysis here; later on, for effective matching, optimal hyperparameters are determined using the grid search (section 6.2.3).

Results As an example of the cross-comparison results, we take a closer look at the following sentence in our subset:

- (1) Ref: **Nun** kann man für eine Person nachschauen.
 DIT: **Jetzt** kann man für eine Person nachschauen.
 DAT: **Nun** kann man für eine Person nachschauen.

Clip-ID: 3635637e8b48ad0fb893718e55d759d838fe42e2a3b5c0a2619ceab31c29155a

Table 5.1 shows the cross-comparison output for the reference word *nun* ($i = 0$) in this sentence for both DIT and DAT. For each pair, the indices i and j , the word forms r_i and h_j , and the three similarities sim_{lex} , sim_{pos} , and sim_{comb} are reported, with the row of the highest sim_{comb} highlighted.

TABLE 5.1: Cross-comparison for reference word *nun* ($i = 0$). Global lexical normalization ($L_{\text{max}}^{\text{word}} = 26$), squared positional similarity deactivated, $\alpha = 0.7$. The row with the highest sim_{comb} is highlighted.

(A) DIT						
i	j	r_i	h_j	sim_{lex}	sim_{pos}	sim_{comb}
0	0	nun	jetzt	0.808	1.000	0.865
0	1	nun	kann	0.885	0.833	0.869
0	2	nun	man	0.923	0.667	0.846
0	3	nun	für	0.885	0.500	0.769
0	4	nun	eine	0.885	0.333	0.719
0	5	nun	person	0.808	0.167	0.615
0	6	nun	nachschauen	0.692	0.000	0.485

(B) DAT						
i	j	r_i	h_j	sim_{lex}	sim_{pos}	sim_{comb}
0	0	nun	nun	1.000	1.000	1.000
0	1	nun	kann	0.885	0.833	0.869
0	2	nun	man	0.923	0.667	0.846
0	3	nun	für	0.885	0.500	0.769
0	4	nun	eine	0.885	0.333	0.719
0	5	nun	person	0.808	0.167	0.615
0	6	nun	nachschauen	0.692	0.000	0.485

Clip Path: datasets/STT4SG-350 v2.1/clips_test/226c7f4c-0da0-4e94-a873-ad1586107355/3635637e8b48ad0fb893718e55d759d838fe42e2a3b5c0a2619ceab31c29155a.mp3

The mean sentence lengths of all 100 samples in the subset of the reference, DIT hypothesis, and DAT hypothesis are reported in Table 5.2.

TABLE 5.2: Mean sentence length (in words) across 100 Wallis samples, for the reference and both transcription systems.

Source	Mean length
Reference	7.89
DIT	8.04
DAT	7.86

Furthermore, the distributions of all three similarities sim_{lex} , sim_{pos} , and sim_{comb} of all 100 samples in the subset are reported in Figure 5.1. On top, the plots of the DIT model can be observed, while the plots at the bottom correspond to the DAT model. For every similarity distribution, the mean is also visible as a dashed line. Only word-to-word pairs are displayed; insertions and deletions carry no similarity in the cross-comparison. The sim_{comb} was computed with global lexical normalization, squared positional similarity deactivated, ($L_{\text{max}}^{\text{word}} = 26$) and $\alpha = 0.7$.

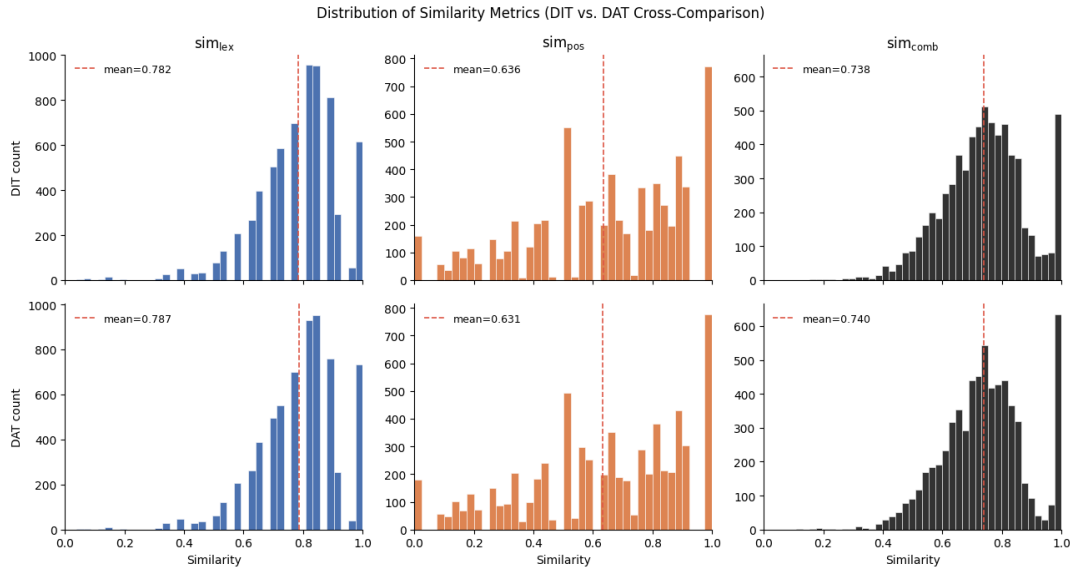


FIGURE 5.1: Distribution of sim_{lex} , sim_{pos} , and sim_{comb} across all word pairs from the cross-comparison on 100 Wallis samples, shown separately for the dialect-ignorant (DIT, top row) and dialect-aware (DAT, bottom row) transcripts. Dashed lines indicate the mean of each distribution. Computed with global lexical normalization ($L_{\text{max}}^{\text{word}} = 26$), squared positional similarity deactivated and $\alpha = 0.7$.

For our example (1), there is a dialectal variation of the reference word *nun*, which has been transcribed in the DIT as *jetzt*. In Table 5.1, we can observe how the similarities behave; as anticipated, because of the global normalization, the sim_{lex} already starts at a rather high value (*nun* $\rightarrow$ *nachschauen* with a sim_{lex} of 0.692), even though they are completely different words. Since, for our purposes, only the order of similarities matters, this is not an issue.

This behavior is also visible for all 100 samples in the distribution plots in Figure 5.1. On the left, sim_{lex} starts around 0.4 and rises steeply, peaking at 0.8–0.9, with a mean of 0.782 (DIT) and 0.787 (DAT). The two means are very close to each other, underlining the limitation mentioned by Berthele and Spangenberg [10] of the DIT Whisper model, which still produces transcripts that are too close to the Standard German reference rather than fully capturing dialect-specific variants. However, as the cross-comparison includes all word pair comparisons, the actual matching cases like *nun* $\rightarrow$ *jetzt* contribute very little to the mean, which also leads to a more similar mean of DIT and DAT. Still, a slightly higher DAT mean can be observed, which aligns with its dialect-aware capability, producing a transcript closer to the reference.

The sim_{pos} distribution, on the other hand, covers the full range $[0, 1]$, but is also skewed towards 1, with a mean of 0.636 (DIT) and 0.631 (DAT), since in an exhaustive pairwise comparison, small position differences occur more often than large

ones (only the corner pairs reach the maximum gap of $\max(|r|, |h|) - 1$). The effect is amplified in longer sentences, which also explains the slightly higher DIT sim_{pos} mean: the mean DIT sentence length is 8.04 — possibly due to dialect-specific insertions — while the DAT mean (7.86) is nearly identical to the reference (7.89), as shown in [Table 5.2](#).

These findings for sim_{lex} and sim_{pos} carry over to the sim_{comb} distribution.

Returning to example (1) and the highlighted row in [Table 5.1](#): a naive matching based on the highest sim_{comb} selects *nun* $\rightarrow$ *kann* in the DIT cross-comparison, even though *kann* clearly belongs to the reference *kann* at the next position. The metric has no way to know that the other words are already accounted for. If we could enforce a 1:1 alignment — each reference word matched to exactly one hypothesis word — the only plausible candidate left for *nun* would be *jetzt*, the correct match. This is exactly the motivation for the **Weighted Bipartite Matching** approach introduced in [chapter 6](#).

Chapter 6

Word Alignment

As shown in [section 5.4](#), simply selecting the hypothesis word with the highest sim_{comb} for each reference word does not guarantee a sensible alignment. What we want instead is a 1:1 matching that finds the best combination of word pairs across the *whole* sentence. This chapter describes how we achieve this using weighted bipartite matching and how its hyperparameters are selected.

6.1 Weighted Bipartite Matching

We formulate our word alignment as a min-cost max-flow problem on a weighted bipartite graph [13], where every reference word is matched to exactly one hypothesis word.

6.1.1 Partitions

The reference and hypothesis words form the two partitions of the graph: R is the set of reference words, and H is the set of hypothesis words. Each word corresponds to a single node in its partition.

As a short visualization, we consider the following example sentence:

- (1) Standard German: Nun esse ich.
 Swiss German: Jetzt esse ich.

The partitions of the corresponding bipartite graph are shown in [Figure 6.1](#).

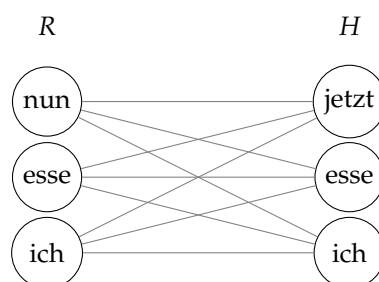


FIGURE 6.1: Bipartite graph for example (1): reference words (R) and hypothesis words (H) form two disjoint partitions, with edges representing all possible pairings.

However, reference and hypothesis sentences also differ in length due to insertions and deletions. Some Standard German words are dropped in the Swiss German dialect, while in other cases, dialect-specific words are added. The following examples, drawn from everyday Swiss German speech, illustrate both phenomena.

- (2) Standard German: Ich gehe — essen.
 Swiss German: Ich gahn **go** esse.

*Example of an insertion: the Swiss German particle **go** has no Standard German counterpart.*

- (3) Standard German: Er fängt an **zu** essen.
 Swiss German: Er fangt ah — esse.

*Example of a deletion: the infinitive marker **zu** is dropped in this Swiss German variant.*

Beyond these linguistic cases, the DIT and DAT transcripts often contain insertions and deletions arising from ASR errors — words missed by the model or extra words inserted from phonetic noise. Because of both of these cases, the bipartite matching must therefore handle unmatched words on both sides.

We address this by introducing ε -nodes to our graph, visualized in Figure 6.2. Reference words from a deletion case are matched with ε -nodes on the hypothesis side, while hypothesis words from an insertion case are matched with ε -nodes on the reference side. In theory, a single deletion or insertion would only require one ε -node. However, we decided to pad each partition with as many ε -nodes as the other partition has real words. This way, we give every word the option to either match with another word or match to an ε -node.

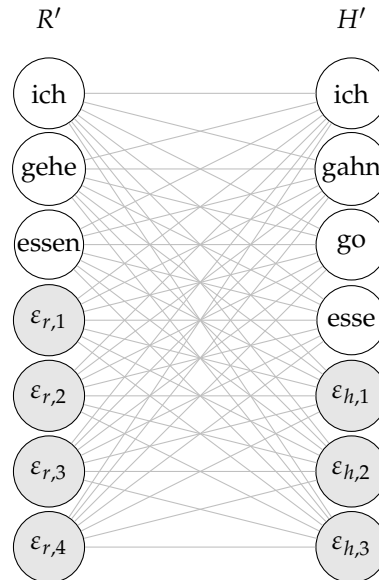


FIGURE 6.2: Augmented bipartite graph for the insertion example in (2). The reference partition R' contains 3 real reference words and 4 ε -nodes (shaded), matching the size of H' . The hypothesis partition H' contains 4 real hypothesis words and 3 ε -nodes (shaded). Both augmented partitions have size $|R'| = |H'| = 7$.

Having defined the partitions, we now turn to the formal network definition in [subsection 6.1.2](#).

6.1.2 Formal Definition

Let $R = \{r_1, \dots, r_{n_r}\}$ be the set of reference words and $H = \{h_1, \dots, h_{n_h}\}$ be the set of hypothesis words, where $n_r = |R|$ and $n_h = |H|$. We define the augmented sets R' and H' with cardinality $N = n_r + n_h$:

- **Reference:** $R' = R \cup \{\varepsilon_{r,1}, \dots, \varepsilon_{r,n_h}\}$
- **Hypothesis:** $H' = H \cup \{\varepsilon_{h,1}, \dots, \varepsilon_{h,n_r}\}$

By construction, $|R'| = |H'| = N$.

The alignment is modeled as finding the min-cost max-flow in a bipartite network $G = (V, E)$ with node set $V = R' \cup H' \cup \{s, t\}$, where s is the source and t the sink. The edge set E contains directed edges from s to every node in R' , from every node in R' to every node in H' (the bipartite layer), and from every node in H' to t .

For the insertion example, the bipartite graph is visualized in [Figure 6.3](#).

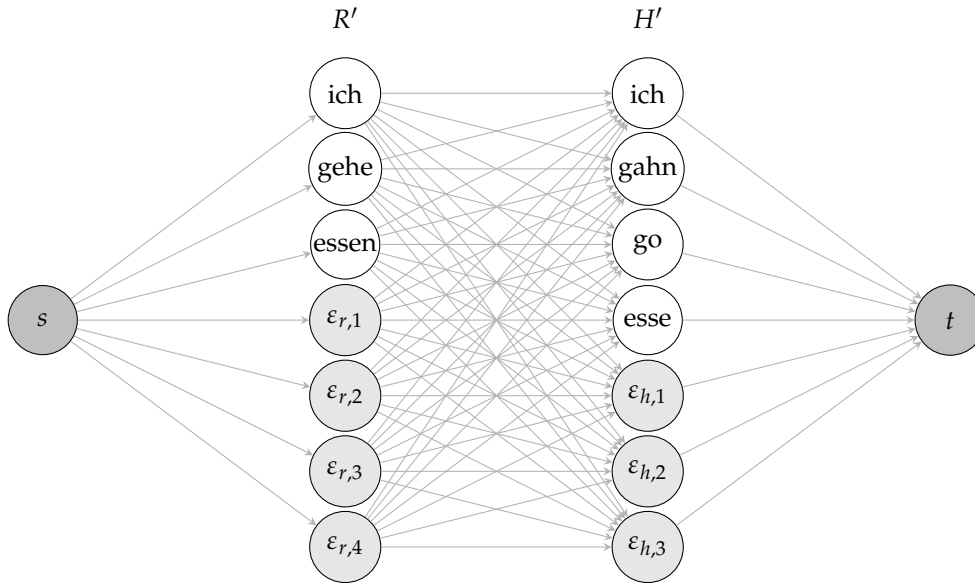


FIGURE 6.3: Full min-cost max-flow network $G = (V, E)$ for the insertion example (2). The source s connects to every node in R' , every node in R' connects to every node in H' , and every node in H' connects to the sink t . ε -nodes are shaded; the source and sink are highlighted in darker gray.

We refer to the algorithm that solves this min-cost max-flow problem as the **solver**.

6.1.3 Cost Functions

Since our bipartite graph is weighted, we now define the edge costs. The cost is what tells the solver which alignments are preferable; lower cost means a better match. We use the combined similarity sim_{comb} introduced in [section 5.3](#).

Because sim_{comb} measures how similar two words are, the cost is the opposite: the further apart two words are, the higher the cost of matching them. We therefore define the cost as 1 minus sim_{comb} :

$$\text{cost}(r_i, h_j) = 1 - \text{sim}_{\text{comb}}(r_i, h_j) \quad (6.1)$$

For ε -edges (representing deletions and insertions), there is no similarity to compute. We therefore introduce λ as a **penalty** for matchings involving ε -nodes. Since the solver minimizes total cost across the whole sentence, λ influences how willing the solver is to route a word through ε rather than match it to a real word: a high λ makes ε -matching expensive, encouraging the solver to accept even low-similarity real matches; a low λ makes ε -matching cheap, so the solver tends to route words through ε instead of matching them to a poor real counterpart. λ is a hyperparameter determined in the grid search (section 6.2.3).

Furthermore, $\varepsilon \rightarrow \varepsilon$ matchings receive a cost of 0 since they are not part of the actual word alignment. Additionally, all edges from the source s to R' and from H' to the sink t carry a cost of 0, as they represent flow infrastructure rather than actual matching decisions and, therefore, should not contribute to the optimization objective.

The cost c_{ij} for matching an element $i \in R'$ with an element $j \in H'$ is defined as:

$$c_{ij} = \begin{cases} \text{cost}(r_i, h_j) & \text{if } r_i \in R, h_j \in H \\ \lambda & \text{if } (r_i \in R, h_j \notin H) \vee (r_i \notin R, h_j \in H) \\ 0 & \text{if } r_i \notin R, h_j \notin H \end{cases} \quad (6.2)$$

Analogously, the total cost is defined as the sum of all c_{ij} over the edges carrying flow and therefore being part of the alignment:

$$C = \sum_{(i,j) \in E} c_{ij} \cdot f_{ij}. \quad (6.3)$$

6.1.4 Optimization

With the cost functions defined, the solver finds the flow assignment f that minimizes the total cost C , subject to the following constraints [13].

Constraints To enforce a 1:1 matching, each edge carries either 0 or 1 unit of flow, with edges carrying flow 1 representing the actual alignments. A feasible assignment satisfies:

- **Edge capacity:** each edge carries at most one unit of flow,

$$f_{ij} \in \{0, 1\} \quad \forall (i, j) \in E. \quad (6.4)$$

- **Flow conservation:** for every interior node, incoming flow equals outgoing flow,

$$\sum_{i:(i,k) \in E} f_{ik} = \sum_{j:(k,j) \in E} f_{kj} \quad \forall k \in V \setminus \{s, t\}. \quad (6.5)$$

- **Total flow:** the source emits N units of flow (equal to the augmented partition size), and by conservation, the sink absorbs the same amount,

$$\sum_{j:(s,j) \in E} f_{sj} = \sum_{i:(i,t) \in E} f_{it} = N. \quad (6.6)$$

Objective The solver minimizes the total cost subject to these constraints:

$$\min_f C. \quad (6.7)$$

The optimal flow assignment for the example (2) is visualized in Figure 6.4.

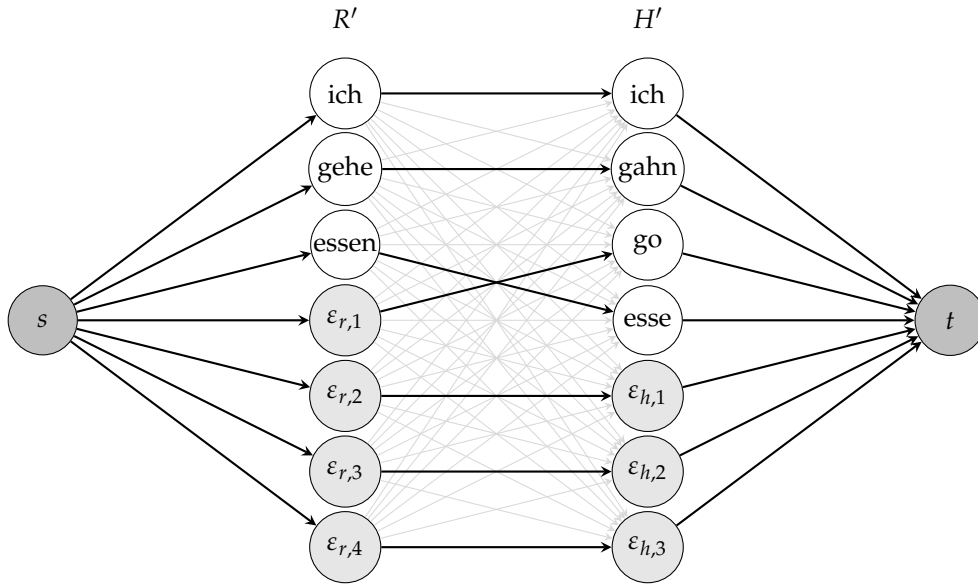


FIGURE 6.4: Optimal flow assignment for the insertion example (2). Edges carrying flow $f_{ij} = 1$ are shown in bold black, indicating the chosen matchings; edges with $f_{ij} = 0$ are shown in light gray. The bipartite layer reveals the alignment: $ich \rightarrow ich$, $gehe \rightarrow gahn$, $essen \rightarrow esse$, and the insertion go is routed through an ϵ -node ($\epsilon_{r,1} \rightarrow go$). Remaining ϵ -nodes match $\epsilon \rightarrow \epsilon$ at zero cost.

6.1.5 Implementation

We implemented the bipartite graph using the NetworkX library [21], specifically the `min_cost_flow()` function, which is based on the network simplex algorithm [13]. After calling `min_cost_flow()`, we structured the output as a Python dictionary mapping reference to hypothesis nodes. Additionally, we implemented a helper function `build_reduced_graph_by_matching()` to retrieve the alignment as a reduced graph M containing only the matched edges.

$$M = \{(r_i, h_j) \in R' \times H' : f_{r_i, h_j} = 1 \wedge \neg(r_i \notin R \wedge h_j \notin H)\}. \quad (6.8)$$

For visualization purposes, M also includes the corresponding $s \rightarrow R'$ and $H' \rightarrow t$ routing edges, enabling the visualization of the alignment as a graph using NetworkX's drawing functions. The results are shown and interpreted in section 8.2.

6.2 Hyperparameters

While our graph G and its min-cost max-flow alignment guarantee that the optimal matching is found for the reference sentence given our total cost function C defined in Equation 6.3 [13], the quality of this matching depends on the chosen hyperparameters. Specifically, we can configure the following options:

- **Lexical positional weighting** $\alpha \in [0, 1]$
- **Insertion/deletion penalty** $\lambda \in [0, 1]$
- **Lexical normalization mode:** `use_global_lexical_normalization`
- **Squared or linear positional similarity:** `use_squared_positional`

In this section, we describe how these hyperparameters are selected.

6.2.1 Test Samples

First, a subset of 100 audio clips from the test set of the STT4SG-350 dataset [3] was defined to evaluate the hyperparameter candidates. The samples were initially drawn with seed=42, which yields a random yet deterministic selection for reproducibility. However, the resulting subset contained a flawed transcription caused by ASR hallucination (*Summe hävera jufgi budd.*), which is not a meaningful Swiss German sentence. Since reliable ground-truth transcripts are essential for hyperparameter selection, the seed was incremented to 43. The selected audio clips are reported in Table A.1.

The goal was to identify hyperparameters that perform robustly across various dialects. Although evaluating every dialect separately would have been ideal, the time-consuming nature of establishing ground truth transcripts necessitated a more targeted approach. Consequently, we limited our evaluation to transcripts from Zurich German and Wallis German. This decision is motivated by the work of Paonessa et al. [7], who find that Wallis German is the most distant of the Swiss German dialects, while Zurich German lies centrally, close to most other regions. By selecting these two varieties, we covered the primary typological extremes of the Swiss German dialect continuum, allowing us to seek a balanced hyperparameter compromise. To validate this approach, the hyperparameters for Zurich and Wallis are also evaluated separately.

Since the test subset of the STT4SG-350 dataset contains recordings of the same reference sentences spoken by Zurich and Wallis speakers [3], the same 100 audio clips were used for both dialects. By keeping the lexical content constant, this ensured that any differences in the hyperparameter evaluation were a direct result of dialectal variation rather than sentence-level differences.

The 100 samples from Wallis are the same as those used in section 5.4.

6.2.2 Ground Truth

The next step was to establish the ground truth. To evaluate the output quality of a solver, we first needed to determine which matching should be regarded as optimal.

For this purpose, we used a small terminal-based application that provided an initial alignment proposal using our bipartite solver with hyperparameters $\alpha=0.7$ and $\lambda=0.3$, with global lexical normalization and squared positional weighting both disabled. These initial values were based on exploratory insights gained during the implementation of the bipartite matching.

```

[69/100]
path: datasets/STT4SG-350 v2.1/clips__test/a1859917-0a13-49e8-aa92-77ac5cb5151e,
 $\alpha=0.7$   $\lambda=0.3$  use_global_lexical_normalization=False

REF: [0]er [1]zog [2]den [3]fall [4]weiter
HYP: [0]er [1]hätte [2]der [3]fall [4]weitergezogen



| ref        | → | hyp                   | similarity |
|------------|---|-----------------------|------------|
| [0] er     | → | [0] er                | 1.000      |
| [1] zog    | → | $\epsilon$ (deletion) | —          |
| [2] den    | → | [2] der               | 0.767      |
| [3] fall   | → | [3] fall              | 1.000      |
| [4] weiter | → | [4] weitergezogen     | 0.742      |



enter=accept '<ref> <hyp>'=reassign '<ref> -'=deletion b=back q=save+quit
>

```

FIGURE 6.5: Screenshot: Terminal Tool for Ground Truth Annotation

Subsequently, for each word in the reference sentence, the reviewer could choose a corresponding word from the hypothesis. To support this process, the similarity score of the suggested alignment was displayed as shown in Figure 6.5.

Although in most cases the ideal matching was intuitive, there were some more ambiguous cases requiring a closer look. For example, consider the following sample:

- (4) Ref: Er **zog** den Fall **weiter**.
Hyp: Er hat den Fall **weitergezogen**.

Clip-ID: 6ab7c208eef2f2d033d50813ddc3f38fc513d5841cb75c57acc95be8fa02386d

Here, in example (4), the sentence has changed because preterite tenses are turned into the perfect in the Swiss German dialect [4], [22]. Furthermore, the verb *ziehen* is combined with the separable prefix *weiter*. Unlike bound prefixes such as *un-* or *ver-*, *weiter* is a free morpheme and can therefore stand alone as an adverb, which enables it to syntactically detach from the verb in the reference (preterite: *zog ... weiter*) but re-attach in the hypothesis (perfect: *weitergezogen*) [23].

We now must decide whether we match *zog* $\rightarrow$ *weitergezogen* or *weiter* $\rightarrow$ *weitergezogen*. Linguistically, *zog* $\rightarrow$ *weitergezogen* would arguably be more interesting, as the underlying verb of *weitergezogen* is *ziehen*. However, as the solver matches based on sim_{comb} , we know that in such cases the separable prefix, such as *weiter*, will dominate, since it is positionally and lexically closer than the verb's preterite form. As it does not make sense to tune hyperparameters against cases the solver cannot structurally solve, we annotate such ground-truth sentences with *weiter* $\rightarrow$ *weitergezogen*.

Besides the separable prefix, however, the auxiliary verb introduced by the perfect transformation, such as *hat* in example (4), is always treated as an insertion during manual ground-truth annotation. In a sentence like *Er zog den Fall.* $\rightarrow$ *Er hat den Fall*

gezogen., such cases are annotated as $zog \rightarrow gezogen$ and $\varepsilon \rightarrow hat$, since the solver can match $zog \rightarrow gezogen$ as they are lexically more similar than $zog \rightarrow hat$ (Levenshtein ratio of 0.6 vs. 0.0 [11]) and are linguistically the more meaningful match.

However, the regional distance in section 8.1 excludes preterite sentences to focus on dialectal variations. Dialectal differences in the preterite-to-perfect shift would be an interesting research topic, but are not covered in this thesis.

In the dialect word lexicon described in section 8.2, however, the preterite cases are included to provide more data. In addition, the ground-truth annotation covers such cases to show that the solver *could* technically handle them through insertions and deletions, and to support both options (preterite included or excluded) in the visualization tool. Tuning the hyperparameters separately for these settings might yield more accurate results; due to time constraints, however, we use the hyperparameters selected from the full subset, even though preterite cases may push the optimal α higher, toward greater weight on lexical similarity, because the perfect construction introduces an auxiliary verb with no direct reference counterpart.

6.2.3 Grid Search

The first approach to finding the optimal hyperparameters is a grid search. The search space is defined as follows:

- $\alpha \in \{0, 0.05, 0.10, \dots, 1.00\}$: 21 evenly spaced values in $[0, 1]$
- $\lambda \in \{0, 0.05, 0.10, \dots, 1.00\}$: 21 evenly spaced values in $[0, 1]$
- `use_global_lexical_normalization` $\in \{\text{True}, \text{False}\}$
- `use_squared_positional` $\in \{\text{True}, \text{False}\}$

This results in a total of $21 \times 21 \times 2 \times 2 = 1,764$ hyperparameter configurations to evaluate.

Procedure For each configuration we build the graph G as defined in section 6.1 and compute the matching using our bipartite min-cost max-flow solver. Based on this matching, we build a reduced matching graph M_{solver} containing only the matched edges.

Analogously, we build a second reduced matching graph M_{gt} for the ground truth against which we compare our solver. The ground truth is based on a manually annotated alignment that represents the matching we actually want to achieve.

For the reduced graphs M , we retain only the following reference-side matched edges:

- $ref\text{-}word \rightarrow hyp\text{-}word$
- $ref\text{-}word \rightarrow hyp\text{-}\varepsilon$ (deletions)

This means that the matched edges of $ref\text{-}\varepsilon \rightarrow hyp\text{-}word$ (insertions) and $\varepsilon \rightarrow \varepsilon$ are excluded from the reduced graphs M .

Since we evaluate the hypothesis against the reference, also including insertions would double-count wrong matches: with a 1-to-1 alignment, a wrong insertion is necessarily paired with either a wrong deletion or a wrong word-to-word match on the reference side. Also, the $\varepsilon \rightarrow \varepsilon$ edges, while structurally required by the min-cost max-flow solver, carry no information about the quality of the word alignment and are therefore also excluded from the evaluation.

With the reduced graphs M_{solver} and M_{gt} , we can now compute the accuracy:

$$\text{accuracy} = \frac{|E(M_{\text{solver}}) \cap E(M_{\text{gt}})|}{|E(M_{\text{gt}})|} \quad (6.9)$$

The metric is bounded in $[0, 1]$, with 1 indicating perfect agreement.

For each sample (a reference/hypothesis sentence pair) in the subset, we compute the accuracy and use the mean across all samples as the final value for the selected configuration:

$$\overline{\text{accuracy}} = \frac{1}{|T|} \sum_{(r,h) \in T} \text{accuracy}(r,h) \quad (6.10)$$

where T is the set of reference/hypothesis sample pairs in the subset.

Results The results of the grid search are reported in Table 6.1.

TABLE 6.1: Grid search results across all configurations. `lex` denotes `use_global_lexical_normalization` and `pos` denotes `use_squared_positional`. The *Ties* column gives the number of grid points achieving the maximum accuracy. The best configuration(s) per section are highlighted in **bold**.

Section	lex	pos	α^*	λ^*	Accuracy	Ties
Wallis DIT	False	False	0.60	0.35	0.96215	1
	False	True	0.75	0.40	0.96477	1
	True	False	0.85	0.20	0.95574	17
	True	True	0.90	0.25	0.95756	16
Wallis DAT	False	False	0.85	0.45	0.99667	1
	False	True	0.85	0.45	0.99667	1
	True	False	0.95	0.30	0.98787	15
	True	True	0.90	0.30	0.98831	15
Zurich DIT	False	False	0.80	0.45	0.98561	2
	False	True	0.85	0.45	0.98561	1
	True	False	0.90	0.20	0.97294	17
	True	True	0.95	0.15	0.97072	1
Zurich DAT	False	False	0.95	0.50	0.98992	11
	False	True	0.95	0.50	0.98992	11
	True	False	0.90	0.30	0.97992	15
	True	True	0.85	0.25	0.98100	16

For each configuration, the best mean accuracy across the 100 samples is reported, along with the corresponding optimal α^* and λ^* values. When several grid points achieve this maximum (a plateau, counted in the *Ties* column), the lowest α^* and λ^* among them are reported. The accuracies show that local lexical normalization consistently performs better than global normalization across all selected sections (Zurich DIT/DAT and Wallis DIT/DAT). The same applies to the squared positional setting, which performs better for Wallis DIT and yields the same accuracy for Wallis DAT, Zurich DIT, and Zurich DAT. Based on these findings, we fix the lexical normalization to local and the positional similarity to squared. This leaves us with the parameters displayed in Table 6.2.

TABLE 6.2: Grid search per-section optima of the word-alignment hyperparameters at the fixed modes.

Section	α^*	λ^*	Accuracy
Wallis DIT	0.75	0.40	0.96477
Wallis DAT	0.85	0.45	0.99667
Zurich DIT	0.85	0.45	0.98561
Zurich DAT	0.95	0.50	0.98992

Since we decided to use the same hyperparameters for all dialect regions and ASR models, we average the α and λ based on our per-section optima, leaving us with the final selected hyperparameters displayed in Table 6.3.

TABLE 6.3: Selected hyperparameters for the final alignment model.

Parameter	Value
α	0.85
λ	0.45
<code>use_global_lexical_normalization</code>	False
<code>use_squared_positional</code>	True

To justify the use of this single global parameter set instead of section-specific tuning, we evaluate how the selected hyperparameters perform against the per-section best accuracy. As reported in Table 6.4, this decision leads to a mean accuracy sacrifice (Δ) of 0.00299, confirming that the unified parameter set is sufficient for our further analysis.

TABLE 6.4: Per-section accuracy at the selected hyperparameters ($\alpha = 0.85$, $\lambda = 0.45$) compared to the per-section optima. Δ represents the accuracy sacrifice incurred by using the unified configuration.

Section	Accuracy	Best	Δ
Wallis DIT	0.95408	0.96477	0.01069
Wallis DAT	0.99667	0.99667	0.00000
Zurich DIT	0.98561	0.98561	0.00000
Zurich DAT	0.98867	0.98992	0.00125
Mean Δ			0.00299
Max Δ			0.01069

Therefore, our selected hyperparameters are $\alpha = 0.85$ and $\lambda = 0.45$, with global lexical normalization disabled and squared positional weighting enabled. In addition, the consistently high accuracies obtained (above 95% across all sections) suggest that the bipartite matching approach is performing reliably overall.

6.2.4 Analytical Approach

As a second approach, we briefly outline an alternative method for finding the optimal values of α and λ analytically.¹ Due to time constraints, this approach is presented as a proposal only and was not implemented or evaluated. Whereas the grid search in [subsection 6.2.3](#) evaluated every parameter configuration, we now take the opposite route: instead of searching over values, we use the ground truth to derive the optimal hyperparameters analytically.

Recall that the solver selects the matching of minimum total cost. As α and λ vary, the optimal matching switches at certain critical values, so the resulting cost is piecewise linear and not everywhere differentiable. Fixing the matching to the ground truth M_{gt} , however, removes this problem. Furthermore, with the word pairs fixed, the lexical and positional similarities sim_{lex} and sim_{pos} are constants for each pair. The quantities behind them, such as the Levenshtein distance, are computed once and no longer vary, so the cost reduces to a function of α and λ alone, which is differentiable everywhere.

However, minimizing this cost alone is not enough: it would only reward making the correct edges cheap, without preventing incorrect edges from becoming equally cheap. We therefore introduce a least-squares proxy F , which also incorporates edges outside M_{gt} , so that the optimal hyperparameters truly separate correct from incorrect edges. The optimal (α^*, λ^*) are the values that minimize F :

$$\begin{aligned}
 F(\alpha, \lambda) = & \underbrace{\sum_{\substack{(r_i, h_j) \in M_{\text{gt}} \\ h_j \neq \epsilon}} (1 - \text{sim}_{\text{comb}}(r_i, h_j))^2}_{\text{(1) matched word pairs: sim} \rightarrow 1} \\
 & + \underbrace{\sum_{(r_i, \epsilon) \in M_{\text{gt}}} \lambda^2}_{\text{(2) matched deletion: } \lambda \rightarrow 0} \\
 & + \underbrace{\sum_{\substack{(r_i, h_j) \notin M_{\text{gt}} \\ h_j \neq \epsilon}} \text{sim}_{\text{comb}}(r_i, h_j)^2}_{\text{(3) unmatched word pairs: sim} \rightarrow 0} \\
 & + \underbrace{\sum_{(r_i, \epsilon) \notin M_{\text{gt}}} (1 - \lambda)^2}_{\text{(4) unmatched deletion: } \lambda \rightarrow 1}
 \end{aligned} \tag{6.11}$$

Squaring gives the standard least-squares form, penalizing a deviation whether it is positive or negative. This keeps the optimum from pushing the correct edges' costs arbitrarily low.

¹This approach was suggested by Andreas Mohn, whom I thank for the discussion.

Each term is designed such that the optimum drives it toward zero:

- **(1)** For matched word pairs, $\text{sim}_{\text{comb}}(r_i, h_j)$ should be close to 1, so we penalize $(1 - \text{sim}_{\text{comb}})^2$.
- **(2)** For matched ε edges, the penalty λ should be small, so we penalize λ^2 directly.
- **(3)** For unmatched word pairs, $\text{sim}_{\text{comb}}(r_i, h_j)$ should be small, so we penalize $(\text{sim}_{\text{comb}})^2$ directly.
- **(4)** For unmatched ε edges, λ should be close to 1, so we penalize $(1 - \lambda)^2$.

Since F is a quadratic polynomial in α and λ , its partial derivatives are linear. Setting them to zero yields a closed-form solution for the optimal hyperparameters:

$$\frac{\partial F}{\partial \alpha} = 0 \implies \alpha^* \quad (6.12)$$

$$\frac{\partial F}{\partial \lambda} = 0 \implies \lambda^* \quad (6.13)$$

The empirical evaluation of this approach, including a comparison of the derived α^* and λ^* to the grid-search results, is left for future work.

Chapter 7

Aligning the Dataset

With the word alignment method defined and its hyperparameters selected in [chapter 6](#), the solver is now fixed and can be applied at scale. This chapter describes how we ran the calibrated solver across the STT4SG-350 dataset [3] and stored the resulting alignments, which form the basis of the dialect analysis in [chapter 8](#).

7.1 Data

For the dialect analysis in [chapter 8](#), we use the DIT and DAT transcripts of the `train_all` subset of the STT4SG-350 [3]. While the full subset was transcribed for completeness, the dataset alignment was restricted to clips marked as usable by the `clip_is_usable` annotation introduced by Berthele and Spangenberg [10]. This annotation excludes silent recordings, clips with unrealistically high speech rates, and clips that could not be reliably transcribed, each manually verified and labeled by them with a corresponding `drop_reason` [10]. Restricting the alignment to usable clips ensures the solver operates only on reliable reference/hypothesis pairs.

7.2 Preprocessing

Before the alignment, we preprocessed the data to normalize the transcripts and remove characters that are not of interest or that could diminish the alignment quality. Otherwise, a word such as *Franken* and its transcribed short form *Fr.* would (a) be harder for the solver to align since the two forms are lexically dissimilar and (b) inflate the dialect word lexicon with such variants.

Concretely, each word was lowercased, and the character β was replaced by *ss*, following the Swiss German convention. Spelled-out units, currencies, magnitude terms, and other frequently abbreviated terms were also contracted to their standard abbreviations. Likewise, the symbol % was replaced by *prozent*. A complete list of the normalized terms is provided in [Table 7.1](#).

Furthermore, like Berthele and Spangenberg [10], we turned numbers into digits using `text_to_num`'s `alpha2digit` function. Unlike `text2num`, which raises an error if its input does not consist entirely of a number word, `alpha2digit` converts the spelled-out numbers it finds within a text and leaves all other words unchanged [24]. Since our preprocessing applies this conversion to every word, this pass-through behavior is necessary.

TABLE 7.1: Fixed normalizations applied to each word during pre-processing. Spelled-out units, currencies, magnitudes, and other frequently abbreviated terms are contracted to their abbreviations; the character β is replaced by *ss* and the symbol % by its written-out form *prozent*.

Original	Normalized form
ß	ss
%	prozent
zentimeter	cm
millimeter	mm
kilometer	km
kilogramm	kg
quadratmeter	m ²
kubikmeter	m ³
milligramm	mg
milliliter	ml
millionen	mio
milliarden	mrd
franken	fr
beziehungsweise	bzw

7.3 Alignment

With the data preprocessed, we ran the solver for every reference sentence against both the DIT and the DAT transcripts, storing the results in two separate Parquet files. Each row represents one word of the reference sentence together with its aligned hypothesis word, as illustrated in Table 7.2.

TABLE 7.2: Example of a stored DIT alignment for one reference sentence. Each row maps a reference word to its aligned hypothesis word, with their sentence indices and the combined similarity. An empty hypothesis word denotes a deletion, where the reference word is matched to an ε -node; an empty reference word represents an insertion. The *path* column is truncated for readability. The DAT alignments follow the same format.

Path	Ref. word	Hyp. word	<i>i</i>	<i>j</i>	Similarity
00e9d8d4...	die	—	0	—	—
00e9d8d4...	zensur	zensur	1	0	0.9542
00e9d8d4...	in	in	2	1	0.9542
00e9d8d4...	den	der	3	2	0.6708
00e9d8d4...	köpfen	köpfe	4	3	0.8769
00e9d8d4...	hat	hat	5	4	0.9542
00e9d8d4...	begonnen	angefangen	6	5	0.4819

Additionally, the DAT and DIT transcripts were aligned directly against each other, with the DAT serving as the reference. This allows us to observe how the approach performs in a setting where no reference sentence is available, relying only on the two ASR outputs.

7.3.1 Implementation

We parallelized the alignment run across multiple worker processes, distributing the sentence pairs into batches. This led to substantial performance increases, resulting in a total runtime of 119.4 s on 15 workers on a MacBook Pro with an M5 Pro chip.

After restricting the data to usable clips, the run aligned 198,274 sentence pairs, producing three alignment files in Parquet format:

- reference → DAT: 1,710,686 alignments (15 MB)
- reference → DIT: 1,746,531 alignments (16 MB)
- DAT → DIT: 1,740,177 alignments (16 MB)

Chapter 8

Dialect Analysis

Having our sentences aligned, we now take a closer look at these alignments.

With our approach, we expect the dialect-ignorant (DIT) model to generate phonetic attempts for words that are specific to Swiss German dialects. At the same time, the parts of the sentence that are close to Standard German should largely be preserved in the hypothesis. Therefore, the incorrectly transcribed DIT words are of primary interest to us; they are the *signal* indicating dialect-specific variation, referred to as DIT variants.

The dialect-aware (DAT) model, on the other hand, is trained on Swiss German and is thus expected to map spoken Swiss German back to Standard German, producing a hypothesis that closely matches the reference. Where the DIT carries the dialectal signal, the DAT stays close to Standard German and instead reflects the general transcription behavior shared by both models. Comparing the two, therefore, allows us to treat the DAT as a *baseline* and isolate the dialect signal in the DIT.

8.1 Regional Distance

As the first step of our dialect analysis, we want to determine how close each dialect region is to Standard German based on the dialectal variations in the sentences.

8.1.1 Measuring Distance to Standard German

Concretely, we do this by comparing the reference separately to the DIT as well as to the DAT, using our bipartite solver. Having these two alignments, we can now compare how well each one matches the reference.

A sentence close to Standard German produces high word similarities and, therefore, a low total cost C , while a sentence with strong dialectal variation produces low similarities and a correspondingly high cost. Since the DAT cost reflects the shared transcription behavior rather than dialectal variation, subtracting it from the DIT cost leaves the additional cost caused by the dialect. By normalizing this cost difference based on the number of reference words $|r|$, we obtain a comparable value for each reference sentence r in the set of analyzed sentences T :

$$\Delta(r) = \frac{C_{\text{DIT}}(r, h_{\text{DIT}}) - C_{\text{DAT}}(r, h_{\text{DAT}})}{|r|}, \quad (8.1)$$

where $C_{\text{DIT}}(r, h_{\text{DIT}})$ and $C_{\text{DAT}}(r, h_{\text{DAT}})$ are the total alignment costs of the reference r against its DIT and DAT hypotheses h_{DIT} and h_{DAT} , and $|r|$ is the number of reference words. As a result, the mean of all the normalized cost differences in a dialect region is the metric we use for measuring how far each dialect region is from Standard German:

$$\Delta_{\text{region}} = \frac{1}{|T_{\text{region}}|} \sum_{r \in T_{\text{region}}} \Delta(r), \quad (8.2)$$

where $T_{\text{region}} \subseteq T$ denotes the set of reference sentences in that dialect region. For the regional distance analysis, we use the `train_balanced` subset of the STT4SG-350 dataset, containing roughly 25,000 sentences per region, or about 17,000 after excluding the preterite cases [3]. Swiss German has no preterite (*Präteritumschwund* [22]); such sentences therefore introduce tense differences, and we exclude them to focus on the dialectal variation.

8.1.2 Wallis German: The Most Distant, and Highest, Alemannic

First, we look at the mean total costs per region in Figure 8.1. We can observe that Wallis is the dialect region furthest from Standard German, consistent with the results of Paonessa et al. [7] and Berthele and Spangenberg [10] as well as the classification of Christen [1].

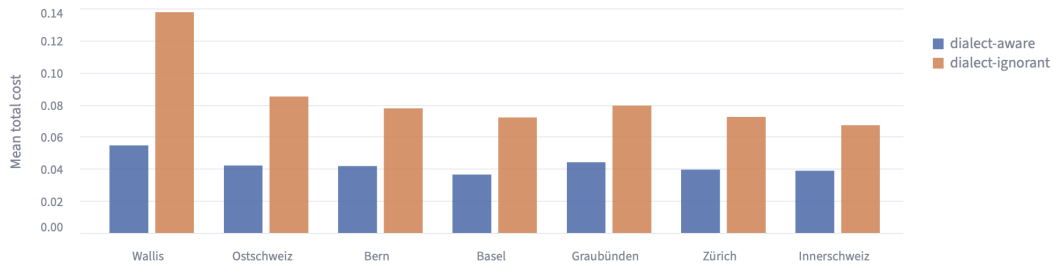


FIGURE 8.1: Mean total alignment cost per dialect region for the dialect-ignorant (DIT) and dialect-aware (DAT) transcripts, each normalized by the number of reference words. A lower cost indicates a closer match to Standard German. Sorted by Δ_{region} .

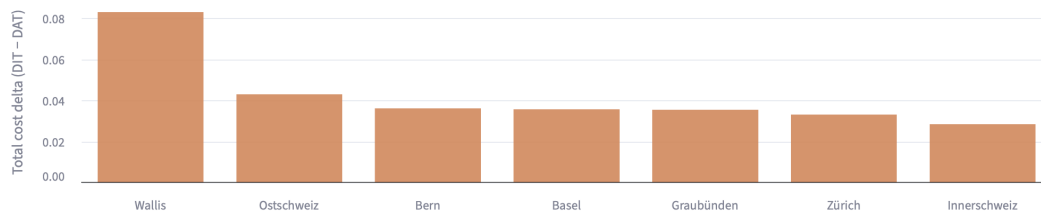


FIGURE 8.2: Regional dialect distance, measured as the mean normalized cost difference Δ_{region} (Equation 8.2) between the DIT and DAT alignments. A higher value indicates a dialect region that is more distant from Standard German. Regions are sorted in descending order.

This pattern is even clearer in the Δ_{region} displayed in Figure 8.2, where Wallis scores nearly twice as high as the second-ranked region, Ostschweiz. At the lower end,

Zurich and Innerschweiz rank second-to-last and last, indicating that they are the closest to Standard German.

However, it is important to note that Graubünden, for instance, carries higher DIT and DAT costs than Basel and Bern, yet is still ranked closer to Standard German when considering only Δ_{region} . This emphasizes the need to consider both views: the separate mean costs in Figure 8.1 as well as Δ_{region} in Figure 8.2. Furthermore, the differences among the remaining dialects — excluding Wallis and Innerschweiz, and arguably Zurich — remain small, making it difficult to draw firm conclusions about their relative closeness to Standard German. The exact figures are reported in Table 8.1. Notably, the DAT cost stays low and relatively stable across all regions (0.0364–0.0547), indicating that the DAT model maps each dialect back closer to Standard German than the DIT does (0.0672–0.1377), confirming our DIT and DAT model selection. However, the DAT cost is still quite high, highlighting the limitation that the DAT model still has potential to be closer to Standard German.

TABLE 8.1: Mean alignment cost per dialect region for the dialect-ignorant (DIT) and dialect-aware (DAT) transcripts, normalized by reference length, with preterite sentences excluded. Δ_{region} (Equation 8.2) is the dialect-distance metric; n is the number of sentences. Sorted by Δ_{region} descending. Table 8.2 reports the same figures including preterite sentences. All values are rounded to four decimal places; Δ_{region} is computed from the unrounded costs and may therefore differ from the difference of the rounded columns by up to 0.0001.

Region	DAT cost	DIT cost	Δ_{region}	n
Wallis	0.0547	0.1377	0.0831	16,951
Ostschweiz	0.0421	0.0851	0.0430	16,959
Bern	0.0417	0.0778	0.0361	16,876
Basel	0.0364	0.0720	0.0356	17,135
Graubünden	0.0441	0.0795	0.0354	16,441
Zurich	0.0395	0.0725	0.0330	16,688
Innerschweiz	0.0388	0.0672	0.0284	17,292

In Table 8.2, the regional distances are reported *including* the preterite sentences. Bern and Graubünden swap ranks compared to Table 8.1, while the remaining regions keep their order. This supports our concern that we cannot draw reliable conclusions about the relative ranking of the middle regions based on this data.

TABLE 8.2: As Table 8.1, but including the preterite sentences. The higher costs and larger n reflect their addition.

Region	DAT cost	DIT cost	Δ_{region}	n
Wallis	0.0581	0.1488	0.0908	24,612
Ostschweiz	0.0458	0.0987	0.0528	24,496
Graubünden	0.0475	0.0940	0.0465	23,939
Basel	0.0398	0.0860	0.0462	24,965
Bern	0.0448	0.0902	0.0454	24,575
Zurich	0.0435	0.0854	0.0420	24,290
Innerschweiz	0.0418	0.0801	0.0383	25,001

8.2 Dialect Word Lexicon

After comparing the dialect regions, we now build a dialect word lexicon. The idea is to extract, from the stored alignments, the aligned word pairs where the hypothesis diverges from the Standard German reference. Under our approach, such divergences potentially carry a dialectal signal. For the lexicon, we use the `train_all` subset of the STT4SG-350 [3], including the preterite sentences, to obtain more data.

8.2.1 Identifying Region-Specific Word Pairs

There are different ways to extract the aligned pairs that carry this signal. One way would be to select the DIT alignments with low similarity. However, this did not yield satisfactory results; it also surfaced incorrectly aligned words and would require choosing a similarity threshold, which is not straightforward to determine, as also reported by Berthele and Spangenberg [10]. A more robust signal is that genuine dialectal pairs tend to appear *often*. For these reasons, we use a count-based approach, specifically **TF-IDF** [20].

TF-IDF is a score that measures how often a certain term appears in a selected document, while rarely occurring in all the other documents. It combines two parts: the term frequency, capturing how often a term appears in the selected document, and the inverse document frequency, capturing how rare it is across all other documents. A document can be any cluster of terms; in the standard case, it is a text such as an article. For example, given a collection of 100 articles, TF-IDF yields a high score for a term that appears often in one article but rarely in all the others [20].

Again, simply applying TF-IDF to the reference or DIT hypothesis words did not yield satisfactory results. We therefore concatenate the reference word and the DIT hypothesis word into a single token, for example *nun+jetzt*, encoding each alignment. We then define each dialect region as a document, yielding seven documents. The TF-IDF score thus rewards tokens that appear often in one dialect region but rarely in all the others. Additionally, alignments where the hypothesis word is identical to the reference, such as *Auto* → *Auto*, carry no dialectal signal and receive a TF-IDF score of zero, so they do not surface among the top pairs. A dialect word specific to a single region, such as the Bernese *Himmugüegeli* for *Marienkäfer*, would receive a high score. As a limitation, genuine dialectal variants shared across all regions will likewise receive a low score and not appear in the lexicon.

8.2.2 Lexicon

With the TF-IDF score, the alignments can now be ranked by how dialect-specific they are. The result is a ranked list of word pairs (reference and DIT variant), of which the 100 highest-scoring are reported in [Table A.2](#). Note that the DIT variant in each pair is Whisper’s Standard German transcription of the spoken form, which may only approximate the actual dialect form. The DIT variant, therefore, signals that a dialectal form occurred but is not itself that form; the dialect forms are reported separately in [Table 8.3](#) for the cases analyzed in detail.

As a short evaluation, each of these 100 pairs was manually annotated by the author, a Swiss German speaker, who listened to the original audio recordings to determine the validity of each pair. A pair is marked as *valid* if it reflects a genuine dialectal correspondence and *invalid* if it does not, for instance, due to an alignment error.

We compared the dialect forms against *idiotikon.ch*, the online *Schweizerisches Idiotikon* [25], a dictionary of Swiss German, as well as the *Dialäktatlas* and the *Kleiner Sprachatlas der deutschen Schweiz* (KSDS) [1], [26], two dialect atlases of German speaking Switzerland. The dictionary documents the forms themselves, while the atlases additionally show their regional distribution.

Valid Alignments For a more detailed analysis, a set of representative cases, including several whose DIT variant is not self-explanatory, was selected and is shown in Table 8.3, divided into three categories: *phonological variants*, *morphological variants*, and *heteronyms (lexical variants)*. For these cases, the actual dialect forms are also reported, showing how the DIT variant relates to the reference word.

The phonological variants include alignments where the word itself remains the same but is pronounced differently. An example is *alle*, which is realized as *au* in Bernese: this reflects *l*-vocalization, a characteristic feature of Bernese German in which syllable-final *l* becomes a vowel-like sound [6]. It is still the same word *alle*, only pronounced differently, which leads Whisper to transcribe it as *auch*. The same process underlies a second case from Bern: *Fall* is pronounced *Fau*, which Whisper transcribes as *v*, the German name of the letter *v* being precisely *Fau*. Another phonological variant is *Oobe*, the Basel German form of *Abend* [25]: the long *a* of *Abend* is rounded to an *o* in many Swiss German dialects [1], [26]. Both the *Dialäktatlas* [26] and KSDS [1] map this variation, recording both *a*-forms (*Aabe*) and *o*-forms (*Oobe*, *Obe*) of *Abend*. The rounded form sounds close enough to the Standard German *oben* that Whisper transcribes it as *such*.

TABLE 8.3: Selected valid dialect word pairs from the TF-IDF analysis, grouped by type of variation. The DIT variant column shows Whisper’s Standard German approximation of the spoken form; the dialect form gives the actual Swiss German realization.

Reference	DIT variant	Dialect form	Peak Region	TF-IDF
<i>Phonological variants</i>				
alle	auch	au	Bern	0.0250
sich	schicht	schich	Wallis	0.0235
abend	oben	oobe	Basel	0.0229
nichts	nix	nix	Wallis	0.0227
hand	hang	hang	Bern	0.0211
gedacht	gedeicht	gedeicht	Wallis	0.0194
prüfung	briefung	priefig	Basel	0.0187
fall	v	fau	Bern	0.0181
gefunden	gefangen	gfunge	Bern	0.0181
erneut	nichts	ernüt	Innerschweiz	0.0169
sollen	hätten	sötted	Innerschweiz	0.0169
<i>Morphological variants</i>				
kätzchen	kätzli	kätzli	Innerschweiz	0.0169
<i>Heteronyms (lexical variants)</i>				
bekommt	kriegt	kriegt	Graubünden	0.0226
geworfen	gerührt	grüehrt	Innerschweiz	0.0169
torhüter	goalies	goalies	Innerschweiz	0.0169

Morphological variants, on the other hand, differ from the reference not in pronunciation but in their morphological structure. The single morphological case in Table 8.3 is the diminutive suffix *-li*: Swiss German forms diminutives with *-li*, while Standard

German uses *-chen*, so *Kätzchen* becomes *kätzli* [6]. The root remains the same; only the diminutive suffix differs.

Lastly, a heteronym is a word that, in another language or dialect system — Swiss German in our case — denotes the same thing as its Standard German counterpart, following Duden (sense 1b) [27]. The use of this term for Standard German–Swiss German correspondences follows Berthele and Spangenberg [10].

In our examples, the Graubünden heteronym *kriegt* corresponds to Standard German *bekommt*. This correspondence is also documented in the *Dialäktatlas* and KSDS [1], [26], which record *kriegen* for *bekommen* in the canton of Graubünden. Similarly, the verb *werfen* corresponds to Swiss German *rüehre*, which surfaces in the data as the participle pair *geworfen* / *gerührt* and is also documented in the Idiotikon and KSDS [1], [25]. *Goalie*, an English loanword used in Swiss German for *Torhüter*, appears here in the plural *Goalies* and therefore likewise acts as a heteronym [28].

Among the 100 alignments in Table A.2, we examine two further cases that do not fit into the three categories introduced above: the *die* → *pro* and *der* → *vor* pairs.

- (1) Reference: er arbeite 70 stunden **die** woche
 Hypothesis (DIT): er arbeitet 70 stunden **pro** woche

Clip-ID: 665854a3bf3c8d916b8bc97a7beaf451456471f24a160c10bf45b16cbe75f3a6

Example of the *die* → *pro* alignment, where the Standard German *die* is replaced by *pro*.

- (2) Reference: das altern beginnt mit dem tag **der** geburt
 Hypothesis (DIT): das altere beginnt mit dem tag **vor** geburt

Clip-ID: 90f99ddf9c4a1ef9af928b1cdf1d6d0ef9dfc9793742cd18362653c6ed5bb7d8

Example of the *der* → *vor* alignment, where the Standard German genitive *der* is replaced by the preposition *vor*.

Let us first take a closer look at example (1). Here, *die* has been changed to *pro*. Both *die Woche* and *pro Woche* are valid German constructions; the speaker used *pro*, whereas the reference transcript has the article *die*, which in Swiss German would merge with the noun (for instance *d'Wuche*) [5], [6]. This is therefore a paraphrase rather than a dialectal form; as the example shows, such a substitution is possible in everyday speech.

In example (2), a case change is visible: *der Geburt* is in the grammatical case *Genitiv* in Standard German and is replaced in Swiss German by the prepositional construction *von der Geburt*, where the noun stands in the *Dativ*. This is common in Swiss German, since the *Genitiv* has mostly been lost (*Genitivschwund*) [6]. This phenomenon also occurs in Standard German, but less commonly [29]. In dialects like Zurich German, this surfaces as two words, *vo de* [5]; however, dialects like Bernese often merge it into a single word, *vor* [6].

These two examples also support our decision to encode each alignment as a single token: had we applied TF-IDF to the reference or hypothesis words alone, the highly frequent words *die* and *der* would have received low scores and never surfaced.

Invalid Alignments Out of the 100 highest-scoring alignments, three were annotated as invalid. These cases are reported in Table 8.4.

TABLE 8.4: The pairs annotated as invalid among the 100 highest-scoring results, with the reason for exclusion.

Pair	Peak Region	TF-IDF	Reason
sei → anscheinend	Graubünden	0.0193	rephrasing
er → heiger	Innerschweiz	0.0192	ASR merge
einer → haben	Bern	0.0181	speaker deviation

To analyze these errors, we look at the examples individually.

- (3) Reference: *Das Problem sei primär die Oberfläche der Venus.*
Hypothesis (DIT): *Das Problem ist anscheinend primär die Oberfläche vor Venus.*

Reference: das problem ε **sei** primär die oberfläche der venus
Hypothesis (DIT): das problem ist **anscheinend** primär die oberfläche vor venus

Clip-ID: dc2d0016f4e24afcd78b573d90ea4de10f71fc43b167d6d76da1e0e4a724fe3a

- (4) Reference: *Die Puppe habe er bereits letzten Herbst zerstört.*
Hypothesis (DIT): *Puppe Heiger bereits im letzten Herbst zerstört.*

Reference: die puppe habe **er** bereits letzten herbst zerstört
Hypothesis (DIT): im puppe ε **heiger** bereits letzten herbst zerstört

Clip-ID: d337f1d1bd13455d0981bb367a65b97a618c2e8bd2ce22a80f5518f6095926f7

- (5) Reference: *Und wieder konnte es einer nicht lassen.*
Hypothesis (DIT): *Und wir haben es nicht lassen können.*

Reference: und wieder konnte es **einer** nicht lassen
Hypothesis (DIT): und wir können es **haben** nicht lassen

Clip-ID: 9d3d6965a0608548b8fc28d60ae6164d329cf06d4024342a97685ef659b997ed

In example (3), the speaker said *ist anscheinend* where the reference has *sei*. This is why the solver matches *sei* to *anscheinend* instead of *ist*, the correct match.

For example (4), the error is due to an ASR merge: the dialectal *habe er* is realized as *heig er*, which the ASR picks up as one word, *heiger*.

In the last error example (5), after manually checking the audio file, we found that the speaker had deviated from the prompt. They said *U mer hei's nid chönne la sii*, which translates to *Und wir haben es nicht sein lassen können*, differing in meaning from the reference. Additionally, the reference uses the preterite (*konnte*), whereas the speaker used a perfect construction due to the *Präteritumschwund* [22]. These differences led to the misalignment of *einer* to *haben*.

8.2.3 DAT as Reference

As a second step, we also extracted the 100 highest-scoring word pairs from the TF-IDF analysis using the DAT transcript as the reference. These results are reported in Table A.3. Unlike Table A.2, these pairs were not manually annotated; the table is provided for comparison with the reference → DIT results.

Of the 100 pairs, 46 also appear in the reference → DIT comparison. This partial overlap shows that the method does not strictly depend on a Standard German reference: it can also be applied with the DAT transcript as the reference, surfacing a related, though not identical, set of dialectal pairs.

8.2.4 Implementation

In the codebase, the TF-IDF score is computed using scikit-learn’s *TfidfVectorizer* [30], [31]. The stored alignments are first filtered to the relevant model and to pairs where both the reference and the hypothesis words are present. Pairs in which the two words are identical carry no dialectal signal and receive a TF-IDF score of zero, so they do not appear among the highest-scoring results.

8.3 Case Studies

Next, we examine two hand-picked cases to showcase our approach in practice. First, we analyze *Butter* → *Anke* across the regions, a commonly used example that appears in the work of Berthele and Spangenberg [10], as well as Brell [12]. The second example is *geweint*, which the DIT transcribes as many different variants and is therefore likewise of interest. Again, we use the `train_all` subset of the STT4SG-350 [3], including the preterite sentences, to obtain more data.

8.3.1 Anke

Anke is a classic example of a dialect-specific word, also documented in the *Dialäktatlas* and KSDS [1], [26]. Our analysis is based on eight alignments across eight sentences.

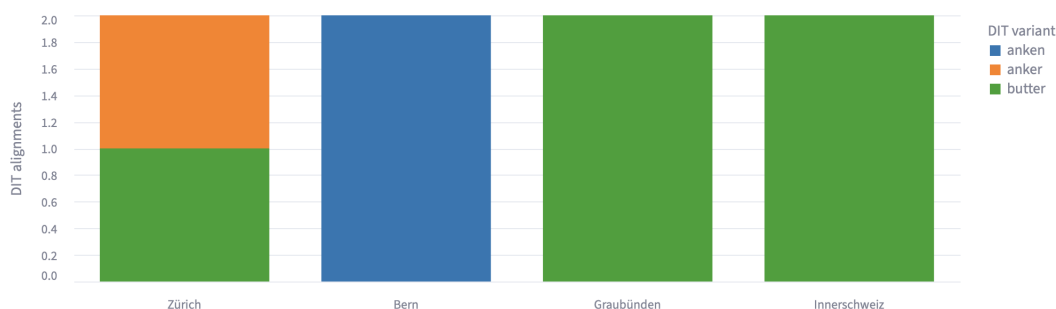


FIGURE 8.3: DIT transcription variants of the reference word *Butter* (dialectally *Anke*) across the four regions in which it occurred. Each region is represented by two alignments.

Figure 8.3 shows that the data for an individual word is sparse: only four of the seven dialect regions are covered, with two alignments each. These results, therefore, serve only as an illustration; to draw firm conclusions, considerably more data would be necessary.

Still, we can observe that the Bernese alignments, together with one from Zurich, use *Anke* or a related form, while the remaining alignments — one from Zurich, two from Graubünden, and two from Innerschweiz — use the Standard German *Butter*. The exact numbers, including the TF-IDF scores, are shown in Table 8.5. The use of *Butter* in Graubünden also aligns with the KSDS, which records *Butter* for this canton [1].

TABLE 8.5: DIT transcription variants of the reference word *Butter*, with occurrence count, mean similarity to the reference, and TF-IDF score.

DIT variant	Count	Mean similarity	TF-IDF
anken	2	0.305	0.01282
anker	1	0.419	0.00785
butter	5	0.965	0.00000

Next, we show what the alignment looks like on the bipartite graph. For this, we take a look at the Zurich *Anke* example (6).

(6) **Transcriptions:**

Reference: *Ich habe immer die falsche Butter verwendet.*

Hypothesis (DIT): *Und ich habe immer den falschen Anker verwendet.*

Hypothesis (DAT): *Und ich habe immer den falschen Anker verwendet.*

Word-level alignments:

Reference: ϵ ich habe immer die falsche **butter** verwendet

Hypothesis (DIT): und ich habe immer den falschen **anker** verwendet

Reference: ϵ ich habe immer die falsche **butter** verwendet

Hypothesis (DAT): und ich habe immer den falschen **anker** verwendet

Hypothesis (DAT): und ich habe immer den falschen anker verwendet

Hypothesis (DIT): und ich habe immer den falschen anker verwendet

Clip-ID: 7cafef572199c8d6537d0133453fde0f4be3bb4ee1a3ebc4911d9cf10beb36a5

Alignment of the sentence for all three comparisons. The DIT and DAT transcripts are identical.

First, the speaker inserted an *und*, which was confirmed by listening to the audio recording. The solver correctly handles this and aligns it to an insertion (ϵ).

Furthermore, the DIT and DAT transcripts are identical in this case. This points to a limitation when using DAT as the reference: with identical transcripts, the pair reduces to an identical *anker* $\rightarrow$ *anker* alignment, and the dialectal signal would not surface at all. Still, the large-scale DAT $\rightarrow$ DIT comparison in Table A.3 shows that the approach can work.

Visualization Finally, we show how the bipartite matching looks using the drawing functions of NetworkX [21]. The result can be observed in Figure 8.4.

We can observe how the *und* node is matched to the ϵ -node, representing the insertion, and how the *butter* $\rightarrow$ *anker* matching receives a low similarity score.

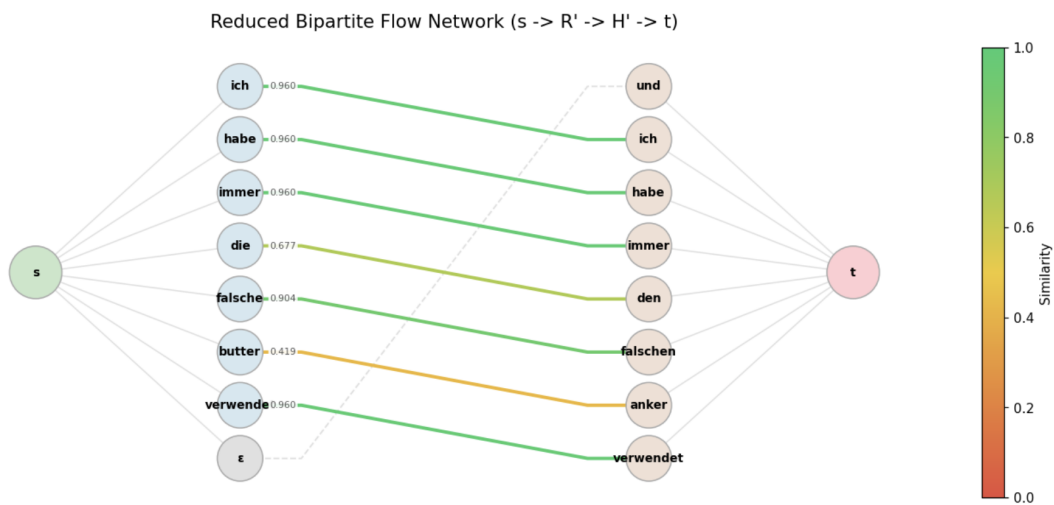


FIGURE 8.4: The reduced bipartite flow network for the alignment of example (6). Reference words (R') on the left are matched to hypothesis words (H') on the right; edge color indicates the similarity score. The reference word *butter* is matched to *anker* at a similarity of 0.419, while *und* is matched to the ϵ -node as an insertion.

8.3.2 Geweint

As the second case study, we analyze the DIT variants of *geweint*. We select the past participle because it produces the clearest dialectal variants: the infinitive *weinen* rarely occurs in the sentences, and the present-tense form *weine* (as in *ich weine*), once lowercased, is indistinguishable from *Weine*, the plural of *Wein* (“wine”).

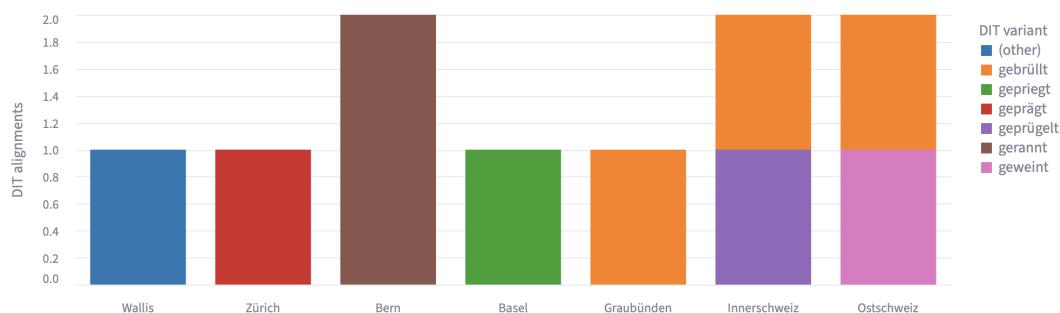


FIGURE 8.5: DIT transcription variants of the reference word *geweint* across the seven dialect regions. The dialectal pronunciations differ enough from Standard German that the dialect-ignorant system transcribes them as a range of distinct words. The *(other)* category groups variants beyond the most frequent ones displayed by the visualization tool; here it contains *geflandert*. With ten alignments in total, the figure remains illustrative rather than statistically representative.

Again, the DIT variants are listed in Table 8.6.

What makes *geweint* instructive is that several of these variants trace back to the *same* spoken dialectal form. When regional pronunciation differs, the DIT model renders

TABLE 8.6: DIT transcription variants of the reference word *geweint*, with occurrence count, mean similarity to the reference, and TF-IDF score.

DIT variant	Count	Mean similarity	TF-IDF
gerannt	2	0.636	0.01282
geprügelt	1	0.575	0.00805
geprägt	1	0.474	0.00785
gepriegt	1	0.568	0.00782
geflandert	1	0.550	0.00515
gebrüllt	3	0.463	0.00508
geweint	1	1.000	0.00000

a single underlying word into several distinct transcriptions. We therefore group the variants by their underlying dialectal form, which we identified by listening to the audio recordings.

brüellet The most common underlying form is *brüellet*, which accounts for the variants in three of the seven regions. It surfaces directly as *gebrüllt* in Graubünden, Ostschweiz, and Innerschweiz. In the two regions with a second speaker, the same form takes a different shape. The other Innerschweiz speaker’s *brüellet* is transcribed as *geprügelt*. In Ostschweiz, the second speaker’s variant is instead picked up as *geweint*; however, after manually checking the audio recording, we found that this speaker also said *brüellet*, aligning with the first speaker. Whisper normalizes it to *geweint*, underlining the limitation that Whisper tends to normalize dialectal speech toward Standard German instead of simply transcribing it.

brieget A second form, *brieget*, surfaces in Zurich and Basel. The underlying word is the same, but the stressed vowel is realized differently in the two dialects, and this difference carries through to the transcription: in Basel, the vowel is closer to an *ie*, which the DIT transcribes as *gepriegt*, whereas in Zurich it sounds more like an *ä*, transcribed as *geprägt*.

grännet In Bern, the dialectal form is *grännet*, which to a model trained on Standard German sounds like *gerannt*. This is the variant with the highest TF-IDF score; it occurs twice, both in Bern.

gflennet Finally, the Wallis variant *geflandert* comes from *gflennet*, the Wallis pronunciation of the Standard German verb *flennen*.

Overall, *geweint* shows how a single dialectal form can fan out into several distinct DIT transcriptions. All the mentioned dialectal forms above are also listed in the KSDS [1]. As before, however, the ten alignments are too few to support claims about which region prefers which variant; they illustrate the phenomenon rather than quantify it.

DAT as Reference Again, we examine how the approach behaves when the DAT serves as the reference. Here, the DAT consistently transcribes the underlying *brüellet* as *geweint*, so it differs from the DIT and the variants *gebrüllt* and *geprügelt* would

still surface. For *geprägt*, *gepriegt*, and *geflandert*, the DAT differs only slightly from the DIT, so these would still provide a signal, though a weaker one than against the Standard German reference. For *gerannt*, the DIT and DAT are identical, so this variant would not surface at all.

8.4 Interactive Tool

For the analysis, we built an interactive tool to visualize the data using Streamlit [32]. In this section, we provide a short overview of the interactive tool, highlighting its core functions.

The tool is divided into two main pages: one for the regional distance analysis presented in section 8.1, and one for the dialect word lexicon discussed in section 8.2. The case studies were produced using the search function of the dialect word lexicon page.

8.4.1 Regional Distance Page

The regional distance page, displayed in Figure 8.6, shows the per-region cost comparison between the DIT and DAT transcripts. Sentences annotated as preterite can be included or excluded; they are excluded for the analysis in section 8.1 to reduce noise.

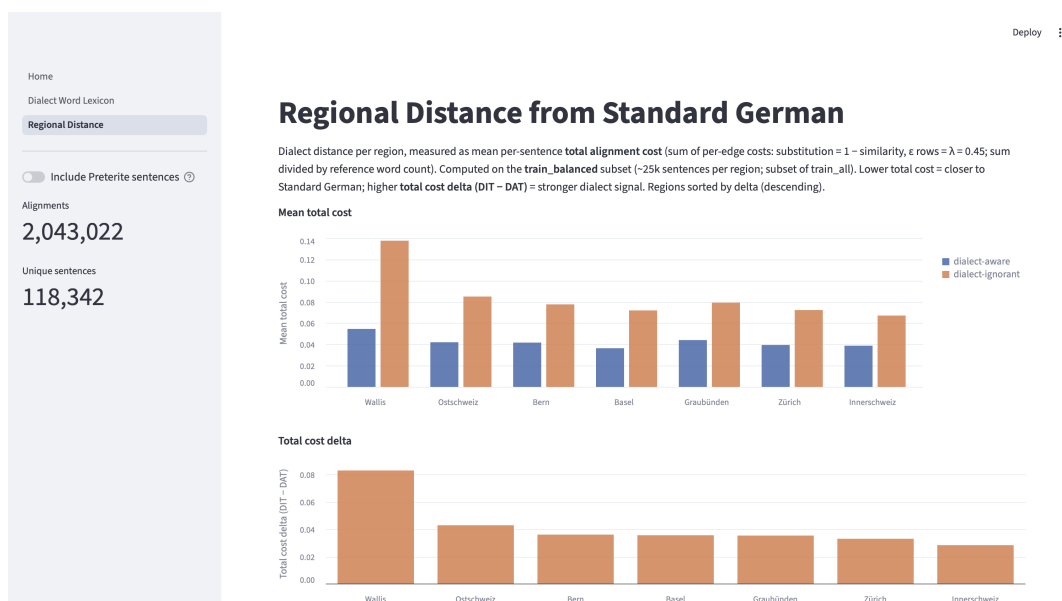


FIGURE 8.6: The Regional Distance page of the interactive Streamlit tool, showing the per-region cost comparison between the DIT and DAT transcripts.

8.4.2 Dialect Word Lexicon Page

The dialect word lexicon page, shown in Figure 8.7, displays a word cloud based on the TF-IDF scores. Below the word cloud, the table of the highest-scoring pairs is reported, matching Table A.2. This view can be switched between Reference $\rightarrow$ DIT and DAT $\rightarrow$ DIT.

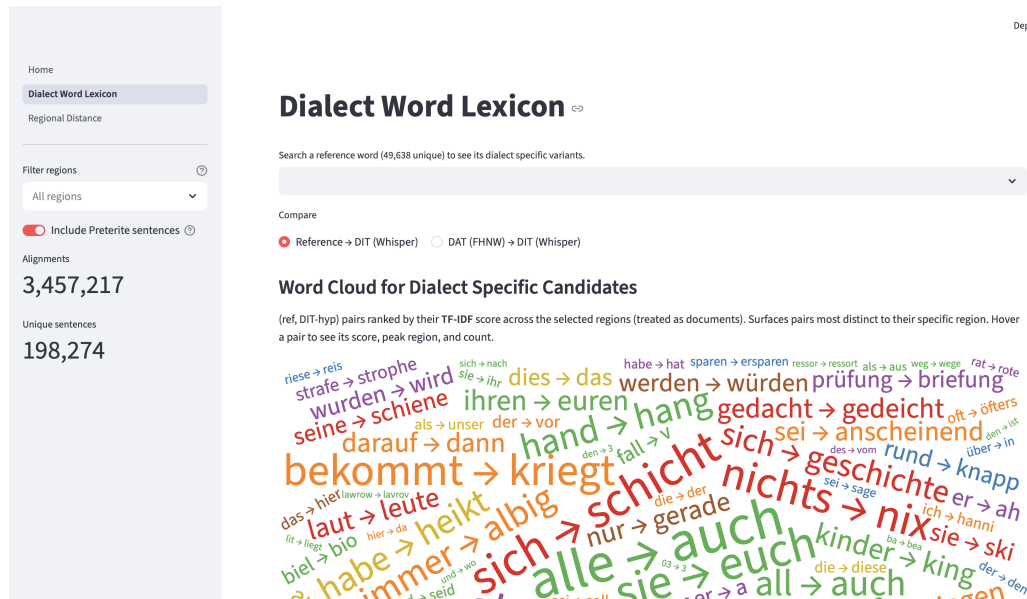


FIGURE 8.7: The Dialect Word Lexicon page of the interactive Streamlit tool. Word pairs are displayed as a word cloud, sized by their TF-IDF score; the user can search a reference word, filter by region, and switch between the Reference → DIT and DAT → DIT comparisons.

8.4.3 Reference Word Search

Furthermore, specific reference words can be searched, as visible in Figure 8.8. The search view displays the DIT variants of the selected word per region. This is the view used for the case studies in section 8.3.

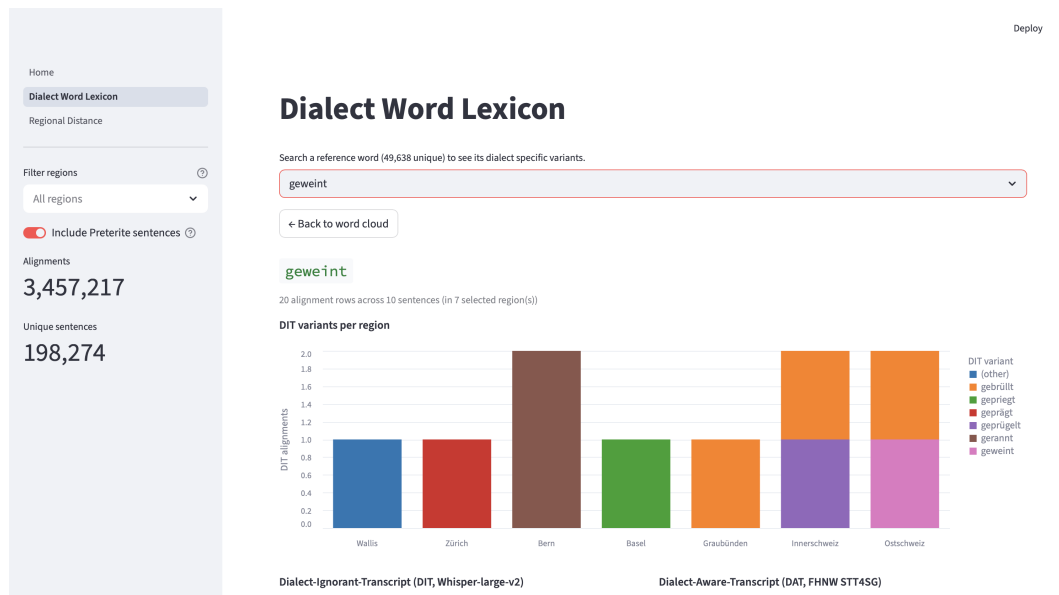


FIGURE 8.8: The search view of the Dialect Word Lexicon page, shown for the reference word *geweint*. Selecting a reference word displays its DIT variants per region.

8.4.4 Reference Word Detail View

Lastly, after scrolling down from the reference word search, the individual sentences and their alignments, grouped by DIT variant, can be explored. The corresponding screenshot is shown in Figure 8.9. This view also includes a function to listen to the audio recordings and to show the technical bipartite alignment (the same visualization used in Figure 8.4).

The screenshot displays the 'Dialect Word Lexicon' interface. On the left sidebar, there are navigation links for 'Home', 'Dialect Word Lexicon', and 'Regional Distance'. Below these, there are filters for 'Filter regions' (set to 'All regions') and 'Include Preterite sentences' (checked). The 'Alignments' section shows '3,457,217' and 'Unique sentences 198,274'. The main content area is titled 'Sentences with word-level alignment: 10 sentences across 7 DIT variants'. It shows a dropdown for 'DIT: gerannt (2)' and a specific clip ID. Below this is an audio player with a 'Load audio' button. The reference sentence is 'Haben Sie auch schon einmal vor einem Patienten geweint.' and the hypothesis is 'Habt ihr auch schon mal vor einem Patienten gerannt?'. The alignment table shows word-level matches between the reference and hypothesis. At the bottom, there is a checkbox for 'Show technical alignment' and a link to the 'Reduced Bipartite Flow Network'.

Reference	haben	sie	auch	schon	einmal	vor	einem	patienten	geweint
Hypothesis DIT	habt	ihr	auch	schon	mal	vor	einem	patienten	gerannt

Reference	haben	sie	auch	schon	einmal	vor	einem	patienten	geweint
Hypothesis DAT	haben	sie	auch	schon	mal	vor	einem	patienten	gerannt

Hypothesis DAT	haben	sie	auch	schon	mal	vor	einem	patienten	gerannt
Hypothesis DIT	habt	ihr	auch	schon	mal	vor	einem	patienten	gerannt

☒ Show technical alignment
REF → DIT: reduced bipartite matching
Reduced Bipartite Flow Network (s → R' → H' → t)

FIGURE 8.9: The detail view of the Dialect Word Lexicon page, reached by scrolling down from the reference-word search shown in Figure 8.8. For each sentence, it displays the reference, DIT, and DAT, the word-level alignments, the audio recording, and the reduced bipartite flow network.

By making the alignment data interactively explorable, the tool allows others to inspect individual dialect words and verify alignments by listening to the audio recordings. Access to the tool is described in Appendix B.

Chapter 9

Discussion and Future Work

9.1 Discussion

The results show that bipartite matching is an effective approach for the word alignment task. Alignment accuracies above 95% against the manually annotated ground truth, obtained in the grid search, confirm this. The chosen similarity measures also proved well-suited: they correctly aligned corresponding words based solely on lexical and positional similarity. This makes the approach lightweight, as it requires no semantic comparison, which would otherwise rely on word embeddings or a trained model. The lexical and positional focus also made the regional distance analysis possible: because the similarity relies on lexical and positional features rather than on meaning, the alignment cost reflects how far a transcription is from the Standard German reference. Wallis showed the greatest regional distance, which aligns with previous findings [7], [10] and supports the validity of the regional distance measure.

The dialect word lexicon analysis was similarly successful: of the 100 highest-scoring pairs, 97 were annotated as valid dialectal correspondences, showing which reference words correspond to dialect-specific variants. These variants also surface linguistic phenomena, allowing phonological, morphological, and lexical variation to be observed. Overall, the TF-IDF approach — encoding the alignments as tokens and treating the dialect regions as documents — produced meaningful results. Furthermore, the case studies illustrated how the DIT variants differ across dialect regions and how this analysis can be conducted using the provided visualization tool.

However, substantial claims regarding which region uses which variant could not be made. This is mainly because there are too few audio recordings for the dialect-specific words. The regional distance analysis could likewise benefit from more data, as only 17,000 sentences per region remained after excluding the preterite cases. One way to address this would be to include the SDS-200 dataset in the analysis [2]. Towards the end of this thesis, we transcribed the `train` split of the SDS-200 with the dialect-aware system, producing the corresponding DAT transcripts; the DIT transcripts were already available from Berthele and Spangenberg [10]. Due to time constraints, however, these results could not be included in the analysis.

Another approach would be to compare the DAT and DIT directly. This removes the need for a reference transcript, with the significant benefit that any spoken Swiss German recording could be processed using the two models alone. The comparison of the two dialect word lexicons, with 46 of the top 100 pairs overlapping, shows that

this approach can work and that meaningful dialectal pairs still surface. However, some interesting cases are buried, as shown in the case studies (section 8.3). The reason is that the two transcripts are not always as cleanly separated as their roles suggest: the DIT, although meant to be dialect-ignorant, occasionally transcribes a dialectal word correctly into Standard German rather than producing a naive approximation, while the DAT does not always map the spoken sentence fully back to Standard German (cf. the *Anke* example in subsection 8.3.1).

9.2 Future Work

Future work, therefore, includes addressing the transcript-model issue: finding a dialect-ignorant model that genuinely does not understand Swiss German and a dialect-aware model that reliably maps it back to Standard German. This limitation was also noted by Berthele and Spangenberg [10]. Resolving it would additionally help the data problem, as it would allow the DIT and DAT to be compared more reliably. A more immediate way to address the limited data is to include the SDS-200 dataset [2] in the analysis. As noted above, the corresponding transcripts were already produced during this thesis, but could not be analyzed in time; incorporating them would increase the number of sentences per region.

The regional distance analysis is currently limited to differences in dialectal words, as the rest of the sentence is expected to be transcribed as Standard German by the DIT. A more comprehensive regional distance measure could also account for finer differences across the entire sentence, such as phonetic variation.

In addition, the hyperparameter selection could be evaluated more thoroughly. The analytical approach in subsection 6.2.4 was only presented as a proposal and not fully applied, and separate, setting-specific hyperparameters — for instance, for the preterite-included and preterite-excluded settings — were not tuned.

Finally, the linguistic analysis could go deeper: differences in insertions and deletions, for example, were not examined in this thesis, although they could reveal further dialectal patterns.

Chapter 10

Conclusion

In this thesis, we followed an ASR-based approach to analyzing Swiss German dialects. The central idea was to contrast a dialect-aware and a dialect-ignorant transcription of the spoken reference text. Where the two differ, the dialect-ignorant model fails to capture a dialectal word and instead produces a phonetic approximation, which serves as a signal of dialectal variation.

To turn this signal into analyzable data, we aligned the reference and hypothesis words using a similarity measure we introduced, combining lexical and positional similarity. Furthermore, formulating the alignment as a bipartite matching problem enabled explicit word-to-word correspondences. We applied the method to the STT4SG-350 corpus, aligning the transcriptions of the full dataset and storing the result as a dataset of reference/hypothesis word pairs.

Based on these alignments, we analyzed Swiss German dialects from several angles. We measured the distance of each dialect region from Standard German and found that Wallis German is the most distant, in line with previous research. We extracted a lexicon of region-specific dialectal word pairs, of which the vast majority represent valid dialectal correspondences. Finally, through case studies of individual words, we showed the different variants that a single Standard German word takes across dialect regions.

At the same time, we faced clear limitations. The two transcriptions are not perfectly separated, as the dialect-ignorant model occasionally normalizes a dialectal word back to Standard German rather than approximating it phonetically. Moreover, the number of recordings per dialectal word is often too small to draw firm conclusions about which region produces which variant. Finally, the analysis so far rests on a single corpus, leaving the SDS-200, which we already transcribed, unexploited.

Beyond Swiss German, the approach is not tied to any particular language. All it relies on is a spoken variety that differs from a written standard, together with dialect-aware and dialect-ignorant transcriptions of the same speech. Future work could therefore include applying this method to another language with a similar dialectal landscape, such as Arabic.

Appendix A

Appendix A: Detailed Results

A.1 Ground-Truth Sample Set

The hyperparameter calibration and cross-comparison are based on a fixed set of 100 samples, each manually annotated to serve as ground truth. [Table A.1](#) lists their clip IDs. The full IDs and associated metadata are available in the repository under:

`experiments/hyperparameter_tuning/ground-truth/samples_metadata.tsv`

TABLE A.1: Clip IDs of the 100 samples used for ground-truth annotation and cross-comparison. IDs truncated to the first 8 hex characters. The same clip ID is used by both the Wallis and Zürich recordings of each sentence; only the speaker (`client_id`) and audio path differ.

#	Clip ID	#	Clip ID	#	Clip ID	#	Clip ID
1	586cd202...	26	e2649fe9...	51	64b3d507...	76	757e1b43...
2	d8730977...	27	3ac54dad...	52	7b4576cf...	77	80c02907...
3	bca8641e...	28	a8d6a2a5...	53	9376a4e8...	78	4438627e...
4	a0210a61...	29	0ed130e5...	54	cf9a5d96...	79	3a93a415...
5	fbf3bcce...	30	6ba66f46...	55	9a856cbb...	80	509860fd...
6	84c4eba3...	31	d55ae1c8...	56	4238e5dd...	81	8e35a5bb...
7	3b0be025...	32	51baf89d...	57	4b91579c...	82	930de885...
8	4b6c1023...	33	cef00772...	58	87216178...	83	b73016a8...
9	79d3fd5c...	34	dade4d77...	59	6babc722...	84	d8f1a675...
10	69227bf0...	35	3f2710fb...	60	70a02a11...	85	6c2a0b9c...
11	d6da60ae...	36	d8514722...	61	82718e89...	86	e6fb3eab...
12	c6fcf570...	37	5e30d695...	62	90ba0573...	87	c53b0e14...
13	f736b396...	38	4f7d31ae...	63	9c156eff...	88	d5ac3fa5...
14	73254ce6...	39	7aff2f9a...	64	3635637e...	89	124988a3...
15	f5782de6...	40	ac525ef4...	65	7d2ee199...	90	c6ac2296...
16	49c84752...	41	37209dac...	66	dc03eb4...	91	43f1781b...
17	d649e2c7...	42	729a17b3...	67	b8bfeb3c...	92	4ce778dd...
18	109d6802...	43	68c55bf1...	68	84511fc3...	93	5d14cd2f...
19	b42155f9...	44	b33d18b9...	69	6ab7c208...	94	cb8f681b...
20	dc099dfb...	45	e5a7db6a...	70	f7c084e9...	95	534d088d...
21	ba72cc5c...	46	94372c3c...	71	ed6cbd65...	96	e2cef433...
22	c4b0ddb7...	47	3adaef1e...	72	efc1d2de...	97	e2ce927a...
23	411f2cca...	48	20d4403c...	73	d6e3b625...	98	3fbdea6d...
24	48eb30e9...	49	ba084179...	74	919760af...	99	e9c5ae0f...
25	e8668c70...	50	8dbbae6f...	75	a6a1c333...	100	3f29eaa3...

A.2 Dialect Word Lexicon

This section presents the full TF-IDF-based dialect word lexicon.

A.2.1 Reference vs. DIT

Table A.2 reports the 100 highest-scoring word pairs from the comparison between the Standard German reference and the dialect-ignorant transcript, which were manually annotated as valid or invalid.

TABLE A.2: The 100 highest-scoring dialect word pairs from the TF-IDF analysis, each manually annotated as valid or invalid based on the original audio recordings. Invalid pairs (e.g. alignment errors or transcription artifacts) are shaded in grey. The count N/M gives the occurrences in the peak region (N) relative to the total across all regions (M).

Pair	Peak region	TF-IDF	Count	Valid
alle → auch	Bern	0.0250	10/10	✓
sich → schicht	Wallis	0.0235	35/35	✓
sie → euch	Bern	0.0234	21/27	✓
rund → gut	Basel	0.0231	7/7	✓
abend → oben	Basel	0.0229	17/18	✓
nichts → nix	Wallis	0.0227	30/30	✓
bekommt → kriegt	Graubünden	0.0226	14/15	✓
ihnen → euch	Bern	0.0225	42/47	✓
immer → albig	Graubünden	0.0212	5/5	✓
hand → hang	Bern	0.0211	6/6	✓
all → auch	Bern	0.0211	6/6	✓
habe → heikt	Innerschweiz	0.0210	5/5	✓
sich → geschichte	Wallis	0.0203	19/19	✓
bekommen → kriegen	Graubünden	0.0198	9/10	✓
kinder → king	Bern	0.0198	5/5	✓
ihren → euren	Bern	0.0197	11/12	✓
gedacht → gedeicht	Wallis	0.0194	16/16	✓
laut → leute	Wallis	0.0194	16/16	✓
nur → gerade	Ostschweiz	0.0194	7/7	✓
ziehen → zeichnen	Graubünden	0.0193	4/4	✓
darauf → dann	Graubünden	0.0193	4/4	✓
ziehen → suchen	Graubünden	0.0193	4/4	✓
sei → anscheinend	Graubünden	0.0193	4/4	–
er → heiger	Innerschweiz	0.0192	4/4	–
nicht → nöd	Zürich	0.0192	9/17	✓
den → der	Graubünden	0.0192	393/2465	✓
sie → schon	Wallis	0.0191	15/15	✓
auch → eure	Innerschweiz	0.0190	8/46	✓
werden → würden	Ostschweiz	0.0190	16/18	✓
sie → ski	Wallis	0.0189	45/46	✓
seine → schiene	Wallis	0.0187	14/14	✓
dies → das	Innerschweiz	0.0187	349/2064	✓
eifersucht → einversucht	Zürich	0.0187	4/4	✓
rund → knapp	Zürich	0.0187	4/4	✓
wurden → wird	Basel	0.0187	4/4	✓
er → ah	Basel	0.0187	4/4	✓
prüfung → briefung	Basel	0.0187	4/4	✓
verstehe → verstand	Basel	0.0187	4/4	✓
vermieden → vermeiden	Basel	0.0187	4/4	✓
prägte → geprägt	Basel	0.0187	4/4	✓
gerät → grad	Basel	0.0187	4/4	✓
auch → oder	Bern	0.0185	9/10	✓
diese → die	Innerschweiz	0.0185	315/1972	✓
er → a	Basel	0.0184	8/9	✓
seiner → verschiedener	Wallis	0.0184	13/13	✓
biel → bio	Bern	0.0181	4/4	✓

continued on next page

Pair	Peak region	TF-IDF	Count	Valid
einer → haben	Bern	0.0181	4/4	–
gefunden → gefangen	Bern	0.0181	4/4	✓
fall → v	Bern	0.0181	4/4	✓
die → pro	Bern	0.0181	4/4	✓
nun → jetzt	Innerschweiz	0.0180	262/1720	✓
der → eines	Wallis	0.0180	12/12	✓
hatte → hat	Zürich	0.0177	283/1657	✓
strafe → strophe	Basel	0.0176	7/8	✓
lösen → lesen	Wallis	0.0175	11/11	✓
aus → unser	Wallis	0.0175	11/11	✓
der → vor	Graubünden	0.0174	42/92	✓
wurde → worden	Graubünden	0.0174	206/1073	✓
lassen → lasse	Bern	0.0173	14/17	✓
partei → partie	Innerschweiz	0.0172	6/21	✓
zahlen → zählen	Ostschweiz	0.0171	5/5	✓
das → hier	Ostschweiz	0.0171	5/5	✓
fahrzeugs → fahrzeug	Ostschweiz	0.0171	5/5	✓
wurden → worden	Graubünden	0.0171	183/1171	✓
die → diese	Innerschweiz	0.0170	188/1268	✓
muss → müsse	Wallis	0.0170	10/10	✓
gehören → kehren	Wallis	0.0170	10/10	✓
diese → tische	Wallis	0.0170	10/10	✓
gute → guter	Graubünden	0.0170	3/3	✓
wehrte → wehrt	Graubünden	0.0170	3/3	✓
oft → etwa	Graubünden	0.0170	3/3	✓
legenden → legende	Graubünden	0.0170	3/3	✓
ich → hanni	Graubünden	0.0170	3/3	✓
sei → soll	Graubünden	0.0170	3/3	✓
oft → öfters	Graubünden	0.0170	3/3	✓
kann → kennt	Graubünden	0.0170	3/3	✓
stande → zustande	Graubünden	0.0170	3/3	✓
entzieht → entzückt	Graubünden	0.0170	3/3	✓
wandte → gewendet	Graubünden	0.0170	3/3	✓
sind → seien	Bern	0.0169	13/16	✓
sind → seid	Bern	0.0169	25/45	✓
trauer → traur	Innerschweiz	0.0169	3/3	✓
aus → ausgestrahlt	Innerschweiz	0.0169	3/3	✓
jugendlicher → jugendlichen	Innerschweiz	0.0169	3/3	✓
torhüter → goalies	Innerschweiz	0.0169	3/3	✓
als → unser	Innerschweiz	0.0169	3/3	✓
geworfen → gerührt	Innerschweiz	0.0169	3/3	✓
letzter → letztes	Innerschweiz	0.0169	3/3	✓
prüfungen → prüfung	Innerschweiz	0.0169	3/3	✓
prozent → 89prozent	Innerschweiz	0.0169	3/3	✓
erneut → nichts	Innerschweiz	0.0169	3/3	✓
fangemeinde → ferngemeinde	Innerschweiz	0.0169	3/3	✓
wie → als	Innerschweiz	0.0169	3/3	✓
kätzchen → kätzli	Innerschweiz	0.0169	3/3	✓
sollen → hätten	Innerschweiz	0.0169	3/3	✓
gar → gerne	Graubünden	0.0168	10/19	✓
die → der	Graubünden	0.0168	163/1073	✓
hatten → habe	Basel	0.0167	6/7	✓
welt → wald	Basel	0.0167	6/11	✓
sprache → spruch	Basel	0.0167	6/7	✓

A.2.2 DAT vs. DIT

The following table Table A.3 reports the 100 highest-scoring word pairs obtained when the DAT is used as the reference. These pairs were not manually annotated and are provided only for comparison with the reference–DIT results above.

TABLE A.3: The 100 highest-scoring word pairs from the TF-IDF analysis using the DAT transcript as reference instead of Standard German. Unlike Table A.2, these pairs were not manually annotated and are provided only for comparison. The count N/M gives the occurrences in the peak region (N) relative to the total across all regions (M).

Pair	Peak region	TF-IDF	Count
sie → ski	Wallis	0.0260	43/43
sich → schicht	Wallis	0.0249	35/35
nichts → nix	Wallis	0.0242	31/31
hand → hang	Bern	0.0230	6/6
lassen → lasse	Bern	0.0229	14/15
habe → heikt	Innerschweiz	0.0229	5/5
abend → oben	Basel	0.0226	13/14
erhält → kriegt	Graubünden	0.0222	19/22
sich → geschichte	Wallis	0.0215	19/19
beziehungen → bezüchungen	Graubünden	0.0214	4/4
ziehen → zeichnen	Graubünden	0.0214	4/4
bekommt → kriegt	Graubünden	0.0212	8/9
den → der	Graubünden	0.0211	367/2265
regnete → geregnet	Innerschweiz	0.0209	4/4
er → heiger	Innerschweiz	0.0209	4/4
auch → oder	Bern	0.0208	10/12
dies → hier	Ostschweiz	0.0207	7/7
nicht → nöd	Zürich	0.0207	9/17
auch → eure	Innerschweiz	0.0206	8/49
gedacht → gedeicht	Wallis	0.0206	16/16
nun → jetzt	Graubünden	0.0202	281/2158
sparen → ersparen	Zürich	0.0202	4/4
will → wollt	Bern	0.0201	9/11
sie → schon	Wallis	0.0199	14/14
seine → schiene	Wallis	0.0199	14/14
gerät → grad	Basel	0.0198	4/4
abends → oben	Basel	0.0198	4/4
verstehe → verstand	Basel	0.0198	4/4
gar → gerne	Graubünden	0.0196	12/24
drückte → ausgedrückt	Bern	0.0196	4/4
biel → bio	Bern	0.0196	4/4
fall → v	Bern	0.0196	4/4
gefunden → gefangen	Bern	0.0196	4/4
wurde → worden	Graubünden	0.0196	227/1194
er → a	Basel	0.0195	8/9
der → eines	Wallis	0.0195	13/13
seiner → verschiedener	Wallis	0.0195	13/13
wurden → worden	Graubünden	0.0194	215/1419
ihren → euren	Bern	0.0194	8/9
werden → kommen	Graubünden	0.0192	6/16
lösen → lesen	Wallis	0.0190	12/12
ihm → nicht	Graubünden	0.0189	3/3
entzieht → entzucht	Graubünden	0.0189	3/3
ich → hanni	Graubünden	0.0189	3/3
coaches → coach	Graubünden	0.0189	3/3
richters → richter	Graubünden	0.0189	3/3

continued on next page

Pair	Peak region	TF-IDF	Count
fdpfraction → fraktion	Graubünden	0.0189	3/3
klären → erklären	Graubünden	0.0189	3/3
die → diese	Innerschweiz	0.0188	207/1271
diese → die	Graubünden	0.0188	175/1236
manche → summe	Wallis	0.0186	11/11
ihr → sie	Bern	0.0185	7/8
auch → oh	Bern	0.0185	7/8
kehrten → sind	Innerschweiz	0.0184	3/3
keines → keiner	Innerschweiz	0.0184	3/3
bestehe → besteht	Innerschweiz	0.0184	3/3
nichts → nie	Innerschweiz	0.0184	3/3
kätzchen → kätzli	Innerschweiz	0.0184	3/3
schlossen → haben	Innerschweiz	0.0184	3/3
aus → ausgestrahlt	Innerschweiz	0.0184	3/3
trauer → traur	Innerschweiz	0.0184	3/3
fangemeinde → fergemeinde	Innerschweiz	0.0184	3/3
auf → aufgewacht	Innerschweiz	0.0184	3/3
prozent → 89prozent	Innerschweiz	0.0184	3/3
strahlte → hat	Innerschweiz	0.0184	3/3
sind → sollen	Ostschweiz	0.0183	5/5
des → vom	Graubünden	0.0183	148/857
deshalb → darum	Innerschweiz	0.0181	159/1094
aus → ausgelesen	Wallis	0.0180	10/10
diese → tische	Wallis	0.0180	10/10
erhielt → gekriegt	Graubünden	0.0180	9/11
beziehung → bezüge	Graubünden	0.0179	5/6
beziehung → bezüchung	Graubünden	0.0179	5/6
bekommen → kriegen	Graubünden	0.0179	5/6
die → der	Graubünden	0.0178	126/914
bands → band	Zürich	0.0178	3/3
gewinnt → geht	Zürich	0.0178	3/3
bettingen → wettingen	Zürich	0.0178	3/3
ums → und	Zürich	0.0178	3/3
fiel → angefallen	Zürich	0.0178	3/3
flüchteten → flüchten	Zürich	0.0178	3/3
sei → sage	Zürich	0.0178	3/3
erregte → erregt	Zürich	0.0178	3/3
gewinnt → gönnt	Zürich	0.0178	3/3
eifersucht → einversucht	Zürich	0.0178	3/3
momentane → momentanen	Zürich	0.0178	3/3
schrei → schreien	Zürich	0.0178	3/3
gefeiert → geführt	Basel	0.0177	6/7
sprache → spruch	Basel	0.0177	6/7
wohl → auch	Bern	0.0176	21/26
der → den	Zürich	0.0176	167/935
menschen → mönche	Bern	0.0176	6/8
gebracht → gebrungen	Innerschweiz	0.0175	5/24
viele → eine	Wallis	0.0175	9/9
ebenfalls → ab	Wallis	0.0175	9/9
selbst → selber	Innerschweiz	0.0175	130/693
sie → ihr	Bern	0.0174	190/414
der → die	Innerschweiz	0.0174	129/857
schriftlich → schriftliche	Basel	0.0174	3/3
schickt → geschickt	Basel	0.0174	3/3

Appendix B

Appendix B: Code Listings and Manuals

B.1 Source Repository

The full source code, artifacts, produced transcripts, and analysis notebooks are available at:

https://github.com/michamettler/BA-FS26_ciel_Mettler_Dialektanalyse

The thesis evaluation refers to commit

29f32b63e9a92285394874070dcf2eb2847d4139.

Set-up instructions for macOS and Windows are provided in the `README.md` at the repository root. Furthermore, access to the interactive web tool described in [section 8.4](#) is also mentioned in the `README.md`.

B.2 Repository Structure

The repository is organized as follows:

```
BA-FS26_ciel_Mettler_Dialektanalyse/
├── datasets/ ..... Input corpora (STT4SG-350, SDS-200)
├── experiments/ ..... Tuning, alignment, and analysis scripts
├── scripts/ ..... Core logic (bipartite matching)
├── transcripts/ ..... Produced DAT and DIT transcripts
├── visualization/ ..... Streamlit tool and figures
├── tests/ ..... Unit tests
├── requirements.txt ..... Dependency list
└── README.md ..... Setup and usage instructions
```

The `experiments/` directory contains the scripts for the various experiments, such as hyperparameter tuning, alignment runs, and cross-comparison, along with the resulting alignment tables stored as Parquet files.

B.3 Key Output Artifacts

The pipeline produces three main types of artifact, all of which are reproducible from the scripts in the repository. The transcripts, both dialect-aware and dialect-ignorant, are stored under

`transcripts/`

The corpus-wide word alignments are stored as three Parquet files under

`experiments/analysis/`

one for each pairing: reference $\leftrightarrow$ DAT, reference $\leftrightarrow$ DIT, and DAT $\leftrightarrow$ DIT. The dialect word lexicon is located under

`experiments/lexicon/`

or can be exported directly from the visualization tool.

Full details on file naming and column structure are provided in the `README.md`.

B.4 Reproducing the Pipeline

The analysis is reproduced in four stages.

1. **Pre-computed transcripts.** Both transcripts (DIT and DAT) are provided with the repository; the subsequent stages consume these TSVs directly.
2. **Calibration.** The calibrated hyperparameters used throughout the thesis are reported in [Table 6.3](#) and are already set in the downstream scripts. The calibration itself can be reproduced by running the grid-search notebook at

`experiments/hyperparameter_tuning/bipartite-matching-hyperparameters.ipynb`

end-to-end; it confirms that the reported configuration is selected against the annotated ground truth.

3. **Alignment tables.** The alignment-table builder at

`experiments/analysis/build_alignment_table.py`

produces, using the calibrated hyperparameters, the three Parquet artifacts: the reference $\leftrightarrow$ DAT, reference $\leftrightarrow$ DIT, and DAT $\leftrightarrow$ DIT alignments for the `train_all` split.

4. **Visualization.** The Streamlit application, launched from its entry point, provides two pages, *Dialect Word Lexicon* and *Regional Distance*, for interactive access to the results. The launch process is described in the `README.md`; the results are based on the alignment tables from step 3.

B.5 Runtime Environment

All experiments were executed on Python 3.12.13 inside a virtual environment managed with `uv`. The package list is provided in `requirements.txt` at the repository root, and the exact versions installed in the evaluation environment are listed in Table B.1.

TABLE B.1: Exact versions of the Python dependencies installed in the evaluation environment. `requirements.txt` at the repository root lists the packages.

Package	Version	Role in the pipeline
pandas	3.0.1	Tabular data manipulation, TSV/Parquet I/O
numpy	2.4.3	Numerical primitives
pyarrow	24.0.0	Parquet read/write for alignment tables
networkx	3.6.1	Bipartite graph construction and min-cost matching
scikit-learn	1.8.0	TF-IDF vectorizer for the dialect lexicon
levenshtein	0.27.3	Lexical similarity (normalized edit distance)
text2num	3.0.1	Word-to-numeral normalization in <code>clean_word</code>
jiwer	4.0.0	Corpus and per-region WER on <code>train_all</code>
openai-whisper	20250625	Whisper large-v2 inference (DIT transcription)
gradio_client	2.3.0	Client for the FHNW STT4SG Gradio API
torch	2.10.0	Whisper backend
torchaudio	2.10.0	Audio I/O for Whisper
transformers	5.3.0	Hugging Face model utilities (Whisper variants)
matplotlib	3.10.8	Bipartite matching plots, grid-search heatmaps
seaborn	0.13.2	Calibration and cross-comparison figures
altair	6.1.0	Streamlit chart rendering
geopandas	1.1.3	Cantonal boundary geometry for the dialect-region map
fiona	1.10.1	GeoPackage I/O (reads <code>swissBOUNDARIES3D</code>)
streamlit	1.57.0	Interactive visualization
streamlit-echarts	0.6.0	Word cloud component
jupyter	1.1.1	Notebook runtime
ipykernel	7.2.0	Notebook kernel
nbformat	5.10.4	Notebook I/O
tqdm	4.67.3	Progress reporting in batch scripts
rich	14.3.3	Console formatting

Bibliography

- [1] H. Christen, E. Glaser, and M. Friedli, Eds., *Kleiner Sprachatlas der deutschen Schweiz*. Frauenfeld: Huber, 2010, ISBN: 9783719315245.
- [2] M. Plüss et al., “SDS-200: A Swiss German speech to Standard German text corpus,” in *Proceedings of the Thirteenth Language Resources and Evaluation Conference (LREC 2022)*, European Language Resources Association, Jun. 2022.
- [3] M. Plüss et al., “STT4SG-350: A speech corpus for all Swiss German dialect regions,” in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, 2023.
- [4] R. Jörg, *Untersuchungen zum Schwund des Präteritums im Schweizerdeutschen* (Basler Studien zur deutschen Sprache und Literatur). Bern: Francke, 1976, vol. 52.
- [5] V. Schobinger, *Zürichdeutsche Kurzgrammatik* (Züritüütsch), 3rd ed. Zürich: Schobinger-Verlag, 2007, ISBN: 9783908105657.
- [6] W. Marti, *Berndeutsch-Grammatik für die heutige Mundart zwischen Thun und Jura*. Bern: Francke, 1985, ISBN: 3772015875.
- [7] C. Paonessa, Y. Schraner, J. Deriu, M. Hürlimann, M. Vogel, and M. Cieliebak, “Dialect transfer for Swiss German speech translation,” in *Findings of the Association for Computational Linguistics: EMNLP 2023*, Association for Computational Linguistics, 2023.
- [8] Y. Scherrer and P. Stoeckle, “A quantitative approach to Swiss German – dialectometric analyses and comparisons of linguistic levels,” *Dialectologia et Geolinguistica*, vol. 24, no. 1, pp. 92–125, 2016. DOI: [10.1515/dialect-2016-0006](https://doi.org/10.1515/dialect-2016-0006).
- [9] R. Hotzenköcherle, N. Bigler, R. Schläpfer, and R. Börlin, *Die Sprachlandschaften der deutschen Schweiz* (Reihe Sprachlandschaft). Aarau: Sauerländer, 1984, vol. 1, ISBN: 3794126238.
- [10] L. Berthele and A. Spangenberg, “Is bern lagging behind zurich? an in-depth analysis of swiss dialects,” Bachelor’s Thesis, Zurich University of Applied Sciences, School of Engineering, Winterthur, Switzerland, Jun. 2024.
- [11] V. I. Levenshtein, “Binary codes capable of correcting deletions, insertions and reversals,” *Soviet Physics Doklady*, vol. 10, no. 8, pp. 707–710, 1966.
- [12] M. Brell, *Recognition of swiss german dialect words based on whisper transcriptions: A case study*, Term paper, Zurich University of Applied Sciences (ZHAW), Feb. 2024.
- [13] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Englewood Cliffs, NJ: Prentice Hall, 1993, ISBN: 0-13-617549-X.
- [14] Federal Office of Topography swisstopo. “swissBOUNDARIES3D.” Wabern, Switzerland, Accessed: May 30, 2026. [Online]. Available: <https://www.swisstopo.admin.ch/de/swissBOUNDARIES3D>

- `www . swisstopo . admin . ch / de / landschaftsmodell - swissboundaries3d`.
- [15] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, *Robust speech recognition via large-scale weak supervision*, 2022. arXiv: 2212.04356 [eess.AS]. [Online]. Available: <https://arxiv.org/abs/2212.04356>.
 - [16] V. Timmel, C. Paonessa, R. Kakooee, M. Vogel, and D. Perruchoud, “Fine-tuning Whisper on low-resource languages for real-world applications,” in *Proceedings of the 10th Swiss Text Analytics Conference (SwissText)*, Association for Computational Linguistics, 2025. [Online]. Available: <https://aclanthology.org/2025.swisstext-1.5/>.
 - [17] i4ds NLP Team, FHNW. “Swiss German Whisper.” Web demo and API for Swiss German speech-to-text, NLP Team at i4ds, FHNW, Accessed: May 28, 2026. [Online]. Available: https://stt4sg.fhnw.ch/long_v3/.
 - [18] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (Prentice Hall Series in Artificial Intelligence). Upper Saddle River, NJ: Prentice Hall, 2000, ISBN: 0130950696.
 - [19] N. Vaessen, *Jiwer: Evaluate your speech-to-text system with similarity measures such as word error rate*, version 4.0.0, Python package, 2024. Accessed: May 28, 2026. [Online]. Available: <https://github.com/jitsi/jiwer>.
 - [20] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge: Cambridge University Press, 2008.
 - [21] A. A. Hagberg, D. A. Schult, and P. J. Swart, “Exploring network structure, dynamics, and function using NetworkX,” in *Proceedings of the 7th Python in Science Conference*, 2008, pp. 11–15.
 - [22] H. Fischer, *Präteritumschwund im Deutschen: Dokumentation und Erklärung eines Verdrängungsprozesses* (Studia Linguistica Germanica). Berlin/Boston: De Gruyter, 2018, vol. 132, ISBN: 978-3-11-056070-1.
 - [23] W. Fleischer and I. Barz, *Wortbildung der deutschen Gegenwartssprache*, 4th ed. Berlin, Boston: De Gruyter, 2012, ISBN: 978-3-11-025665-9. DOI: 10.1515/9783110256659. Accessed: Jun. 4, 2026.
 - [24] Allo-Media. “text2num api documentation.” Version 3.0, Accessed: May 23, 2026. [Online]. Available: <https://text2num.readthedocs.io/en/stable/api.html>.
 - [25] Schweizerisches Idiotikon. “Schweizerisches Idiotikon: Wörterbuch der schweizerdeutschen Sprache,” Accessed: May 25, 2026. [Online]. Available: <https://www.idiotikon.ch>.
 - [26] A. Leemann et al., *Dialektatlas: 1950 bis heute*. Zürich: vdf Hochschulverlag, 2024, ISBN: 978-3-7281-4183-5.
 - [27] Dudenredaktion. “Heteronym, das,” Duden online, Accessed: May 24, 2026. [Online]. Available: <https://www.duden.de/node/66039/revision/1220568>.
 - [28] Dudenredaktion. “Goalie, der,” Duden online, Accessed: Jun. 4, 2026. [Online]. Available: <https://www.duden.de/rechtschreibung/Goalie>.
 - [29] S. Hartmann, *Deutsche Sprachgeschichte: Grundzüge und Methoden*. Stuttgart: UTB, 2018, ISBN: 9783838548234.
 - [30] F. Pedregosa et al., “Scikit-learn: Machine learning in python,” *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011. [Online]. Available: <http://jmlr.org/papers/v12/pedregosa11a.html>.
 - [31] scikit-learn developers. “TfidfVectorizer — scikit-learn documentation,” Accessed: May 25, 2026. [Online]. Available: <https://scikit-learn>.

`org/stable/modules/generated/sklearn.feature_extraction.
text.TfidfVectorizer.html`.

- [32] Streamlit Inc. "Streamlit: A faster way to build and share data apps," Accessed: May 26, 2026. [Online]. Available: <https://streamlit.io>.

Glossary

Bipartite graph

A graph whose nodes can be partitioned into two disjoint sets such that edges only connect nodes from different sets; here, the reference and hypothesis word sets form the two partitions [13].

Genitivschwund

Loss of the genitive case in Swiss German dialects, where genitive constructions are replaced by a prepositional construction with *von* governing the dative (e.g. Standard German *der Geburt* becomes *von der Geburt*, realized as *vo de* in Zurich German or *vor* in Bernese) [6].

Levenshtein distance

The Levenshtein distance between two strings is the minimum number of single-character insertions, deletions, and substitutions required to transform one into the other [11]. In this thesis, it forms the basis of the lexical similarity sim_{lex} .

Min-cost max-flow

A network flow optimization problem that finds a feasible flow of maximum value at minimum total cost; used to compute the optimal word alignments in this thesis [13].

Präteritumschwund

Linguistic phenomenon in Swiss German dialects in which the simple past tense (Präteritum) has been replaced by the perfect tense (Perfekt) [4].

TF-IDF

Term Frequency–Inverse Document Frequency, a statistical measure weighting terms by their frequency within a document relative to a collection; used here to identify regionally distinctive dialect words by treating each region as a document [20].

Wallisertitsch

Swiss German dialect spoken in the canton of Wallis (Valais); classified as Highest Alemannic and the most distinctive of the Swiss German dialects in vocabulary and phonology [1].

WER

Word Error Rate, a standard metric for transcription quality, defined as the sum of word substitutions, deletions, and insertions divided by the number of reference words; based on the Levenshtein distance at the word level [18].