



**Università
di Genova**

DIBRIS DIPARTIMENTO
DI INFORMATICA, BIOINGEGNERIA,
ROBOTICA E INGEGNERIA DEI SISTEMI

Low-resource language translation: A study on Genoese

Matteo Ferrari

Master Thesis

Università di Genova, DIBRIS Via Dodecaneso, 35 16146 Genova, Italy
<https://www.dibris.unige.it/>



MSc Computer Science
Data Science and Engineering Curriculum

Low-resource language translation: A study on Genoese

Matteo Ferrari

Advisors:

Prof. Viviana Mascardi

Prof. Francesca Odone

Dr. Jean Maillard (External)

Examiner:

Prof. Enrico Puppo

October, 2026

Abstract

Neural machine translation for high-resource languages has reached remarkable progress. However, low-resource languages still suffer from severe translation quality bottlenecks due to parallel data scarcity and a lack of standardized writing systems. Genoese, classified by UNESCO as a definitely endangered language, faces both of these challenges. To address this issue, we first developed an orthographic normalization pipeline adopting the *grafia unitäia* endorsed by the Council for Ligurian Linguistic Heritage. Specifically, we trained from scratch an attention-augmented character-level bidirectional Long Short-Term Memory network, outperforming the previous state-of-the-art baseline by reducing the Character Error Rate (CER) from 2.47% to 1.33%. This normalizer was applied to harmonize non-standard data prior to translation training. Four different approaches were explored for the translation phase. Specifically, we employed a baseline trained from scratch by modifying the BiLSTM vocabulary and two sequence-to-sequence models, one supporting Genoese natively (NLLB) and one non-native (mBART), adapted via Quantized Low-Rank Adaptation (QLoRA). Additionally, a decoder-only Large Language Model (EuroLLM) was evaluated, also fine-tuned using QLoRA. The models were evaluated on a custom test set as well as on two global benchmarks, Flores+ and Bouquet. Specifically, on the Bouquet benchmark, they achieved top results in terms of Character F-score plus plus (chrF++) for Italian-to-Genoese translation and the best performance evaluated via MetricX for Genoese-to-Italian translation.

Scintexi

A traduccion automatica neurale a l'ha faeto di gròsci progressi pe-e lengue con de risòrse erte. Con tutto, e varietæ à bassa risòrsa patiscian ancon de gravi problemi de qualitæ apreuvo a-a farta de dæti paralleli e à unna ortografia standardizzâ. O Zeneise, clascificou da l'unesco comme lengua in reisego, o l'affronta tutte doe ste desfie. Pe affrontâ sto problema, emmo primma de tutto sviluppou unna pipeline de normalizzaçion ortografica con adottâ a grafia unitäia approvâ da-o Conseggio pe-o Patrimònio Linguistico Ligure. Ciù into detaggio, emmo addestrou da-o zero unna ræ à memöia longo-curto termine bidirezionale à livello de caratteri e potenziou da l'attençion, passando o passou stato de l'arte e reduxendo o tascio d'errô in scî caratteri (CER) da-o 2,47% à l'1,33%. Sto normalizzatore o l'é stæto applicou pe armonizzâ i dæti no standard avanti de l'addestramento da traduccion. Pe-a fase de traduccion s'é esplorou ciù de unna stradda. Ciù into detaggio, s'é addeuviou unna baseline addestrâ da zero con cangiâ o diçionäio da BiLSTM e doî modelli sequence-to-sequence, un nativo pe-o zeneise (NLLB) e un no nativo (mBART), adattæ con Quantized Low-Rank Adaptation (QLoRA). In ciù, s'é addeuviou un Large Language Model basou solo in sce un decoder (EuroLLM), misso à ponto ascì con QLoRA. I modelli son stæti valuæ in sce un insemme de preuva personalizzæ e in sce doî benchmark globali, Flores+ e Bouquet. Ciù into detaggio, in sciô benchmark Bouquet s'é ottegnuo i megio exiti in ponto Character F-score plus plus (chrF++) pe-a traduccion da l'Italian a-o Zeneise, e e megio prestaçioin valuæ con MetricX pe-a traduccion da-o Zeneise a l'Italian.

This 'Scintexi' was obtained by translating the Italian version of the abstract using the NMT-EuroLLM model explained in Chapter 5

Table of Contents

Chapter 1	Introduction	1
Chapter 2	Background	3
2.1	The Genoese Language	3
2.1.1	Geographical Extension and Historical Context	3
2.1.2	Main Linguistic Concepts	5
2.1.3	Orthography and Standardization Issues	5
2.1.4	Morphosyntactic and Phonetic Divergences	6
2.1.5	Data Scarcity and Digital Presence	7
2.2	Natural Language Processing	7
2.2.1	From One-Hot Encoding to Word Embeddings	8
2.2.2	Tokenization Strategies	9
2.2.3	Back-translation and Data Augmentation	10
2.3	Machine Learning for NLP	10
2.3.1	Encoder-Decoder Architecture	11
2.3.2	Recurrent Neural Networks and the Sequential Paradigm	11
2.3.3	Attention Mechanism	12
2.3.4	The Transformer Architecture	13
2.3.5	Causal Language Modeling	14
2.3.6	Large Language Models	15

Chapter 3	Dataset	16
3.1	Data Sources	16
3.1.1	Online Dictionaries	16
3.1.2	Web Corpora and Digital Outlets	17
3.1.3	Preexisting Parallel Corpora	17
3.1.4	Historical and Heritage Texts	17
3.2	Source Parsing and Format Preprocessing	18
3.2.1	Plain Text Parallel Alignment (<i>O Zinâ</i> and <i>Lunãio</i>)	18
3.2.2	Columnar Data Decompression (ZenaMT)	18
3.2.3	CSV Columnar Extraction (DEIZE)	19
3.2.4	HTML DOM Parsing and Polysemic Expansion (TIG)	19
3.2.5	Monolingual Sentence Extraction (Wikivivagna)	19
3.2.6	Historical Layer Mapping (Casaccia, Piaggio, O Balilla, and Gazzo)	20
3.3	Master Dataset Schema and Aggregation	20
3.3.1	Relational Schema	20
3.3.2	Corpus Statistics and Structure	21
3.4	Dataset Partitioning and Task Alignment	23
Chapter 4	Normalization	24
4.1	Related Works in Text Normalization	24
4.1.1	Evolution of Text Normalization	25
4.1.2	Genoese Orthographic Normalization	26
4.2	Data Processing for Text Normalization	26
4.2.1	Normalization Dataset	27
4.2.2	Dataset Splitting	29
4.3	Methods	30
4.3.1	Task Formulation	31
4.3.2	Norm-RB: Deterministic Rule-Based Extraction	32

4.3.3	Norm-CRF: Conditional Random Fields	33
4.3.4	Norm-BiLSTM: Character-Level Sequence-to-Sequence	35
4.3.5	Data Augmentation Strategies	37
4.4	Experimental Setup	38
4.4.1	Norm-RB: Deterministic Rule-Based Extraction	39
4.4.2	Norm-CRF: Conditional Random Fields	39
4.4.3	Norm-BiLSTM: Character-Level Sequence-to-Sequence	40
4.5	Experimental Results and Discussion	41
4.5.1	Norm-RB: Deterministic Rule-Based Extraction	42
4.5.2	Norm-CRF: Conditional Random Fields	44
4.5.3	Norm-BiLSTM: Character-Level Sequence-to-Sequence	45
4.5.4	Comparative Analysis and State-of-the-Art Benchmark	50
4.5.5	Qualitative Analysis	52
Chapter 5 Translation		55
5.1	Related Works in Machine Translation	55
5.1.1	Multilingual Sequence-to-Sequence Architectures	56
5.1.2	Generative LLMs and Token-Free Models	57
5.1.3	Parameter-Efficient Fine-Tuning Strategies	58
5.1.4	Recent Advances in Low-Resource Language Adaptation	59
5.1.5	Machine Translation for the Genoese Language	60
5.2	Data Processing for Automatic Translation	61
5.2.1	Translation Dataset	61
5.2.2	Dataset Splitting and Dimensions	61
5.3	Methods	62
5.3.1	Task Formulation and Subword Segmentation	63
5.3.2	NMT-BiLSTM: Subword-Level Sequence-to-Sequence	64
5.3.3	NMT-mBART: Unseen Language Adaptation	64

5.3.4	NMT-NLLB: Native Language Adaptation	66
5.3.5	NMT-EuroLLM: Decoder-Only Generative Model	66
5.3.6	Data Augmentation via Back-Translation	67
5.4	Experimental Setup	68
5.4.1	NMT-BiLSTM: Subword-Level Sequence-to-Sequence	68
5.4.2	NMT-mBART: Unseen Language Adaptation	70
5.4.3	NMT-NLLB: Native Language Adaptation	71
5.4.4	NMT-EuroLLM: Decoder-Only Generative Model	72
5.5	Experimental Results and Discussion	75
5.5.1	NMT-BiLSTM: Subword-Level Sequence-to-Sequence	77
5.5.2	NMT-mBART: Unseen Language Adaptation	79
5.5.3	NMT-NLLB: Native Language Adaptation	81
5.5.4	NMT-EuroLLM: Decoder-Only Generative Model	85
5.5.5	Comparative Analysis	88
5.5.6	Qualitative Analysis	94
Chapter 6	Conclusions and Future Work	102
6.1	Summary of Contributions and Main Findings	102
6.2	Future Directions	104
Appendix A	Experimental Setup and Hardware Infrastructure	105
Bibliography		107

Chapter 1

Introduction

The primary objective of this thesis is to address the automated translation from Genoese to Italian and vice versa. Genoese is a prominent regional variety of the Ligurian language family, classified by UNESCO as a definitely endangered language [41]. Working with low-resource languages entails overcoming severe methodological obstacles, including data curation, parallel sentence alignment, and the severe scarcity of digital corpora. Furthermore, Genoese presents an additional layer of linguistic complexity: beyond its limited digital footprint, it lacks a universally codified and standardized writing system. This absence of a single spelling standard leads to high orthographic fragmentation across available texts, where different authors and historical sources employ diverging writing conventions for the exact same words.

To bridge this critical gap, this thesis investigates a comprehensive computational pipeline that treats text normalization as a mandatory prerequisite for translation. By systematically unifying variant spellings into a consistent, rule-governed writing system, the proposed approach harmonizes disparate data sources, thereby expanding and stabilizing the usable parallel corpus for downstream neural translation architectures. Without this preparatory harmonization phase, translation models face severe vocabulary sparsity and structural ambiguity during the learning process.

The remainder of this thesis is structured across distinct phases, detailed as follows:

- **Chapter 2 (Background)** establishes the theoretical and contextual foundation of the work. It provides an in-depth overview of the sociolinguistic history, geographical extension, and morphosyntactic peculiarities of the Genoese language, highlighting the challenges posed by its lack of standardization. Additionally, it reviews the core principles of computational linguistics, the evolution of neural network architectures for sequence processing, and the mechanisms behind subword tokenization and back-translation data augmentation.

- **Chapter 3 (Dataset)** details the extensive data collection and engineering phase. It describes the methodologies employed to harvest, parse, and clean heterogeneous resources, ranging from online dictionaries and web-scraped media to pre-existing parallel corpora and historical heritage texts. It explains how these disparate formats are unified into a single relational master dataset, outlining corpus statistics, data distributions, and the dynamic partitioning strategies tailored for subsequent training tasks.
- **Chapter 4 (Normalization)** focuses entirely on resolving orthographic fragmentation. It formulates text normalization as a character-level sequence-to-sequence transduction task across both word and sentence granularities. The chapter evaluates multiple computational paradigms, including deterministic rule-based extractors, probabilistic sequence labelers, and attention-augmented recurrent neural networks. Furthermore, it introduces length-constrained sequence splitting strategies and back-normalization synthetic data augmentation to overcome data scarcity, culminating in a comparative performance evaluation against established state-of-the-art benchmarks. Related work on text normalization in low-resource settings is discussed directly within this chapter.
- **Chapter 5 (Translation)** presents the core machine translation framework. It explores the transition from character-level mapping to subword-level cross-lingual translation between Italian and Genoese. The chapter evaluates a comprehensive spectrum of architectures, spanning from compact recurrent neural networks trained from scratch to massive multilingual sequence-to-sequence models and generative decoder-only large Language Models adapted via parameter-efficient fine-tuning techniques. It provides an extensive quantitative and qualitative analysis across in-domain and out-of-domain evaluation benchmarks, including official global leaderboard evaluations via the BOUQuET benchmark, discussing the impact of data augmentation and the limitations of automated metrics. Related work concerning machine translation for low-resource languages is integrated within this chapter.
- **Chapter 6 (Conclusions and Future Work)** summarizes the primary contributions and empirical findings of the thesis, reflects on the systemic challenges encountered throughout the experimental phases, and outlines promising avenues for future research in low-resource language processing and evaluation methodologies.

Chapter 2

Background

2.1 The Genoese Language

Genoese (*Zeneize*) is the main variety of the Ligurian language [65], a Gallo-Italic language spoken primarily in the territories of the former Republic of Genoa [66]. While often colloquially referred to as a ‘dialect’ of Italian, it is linguistically distinct, belonging to a different branch of the Romance languages.

2.1.1 Geographical Extension and Historical Context

The linguistic area of Genoese extends throughout the Liguria region in Northern Italy, reaching into bordering areas of Piedmont, Lombardy, and Emilia-Romagna. Historically, as illustrated in Figure 2.1, the maritime expansion of the Republic of Genoa facilitated the spread of the language across the Mediterranean. This diaspora is still evident in linguistic islands such as the Tabarchino variety spoken in Sardinia (Carloforte and Calasetta) and Monégasque in the Principality of Monaco. Furthermore, significant traces of Genoese influence remain in the dialects of Corsica and parts of Southern France.

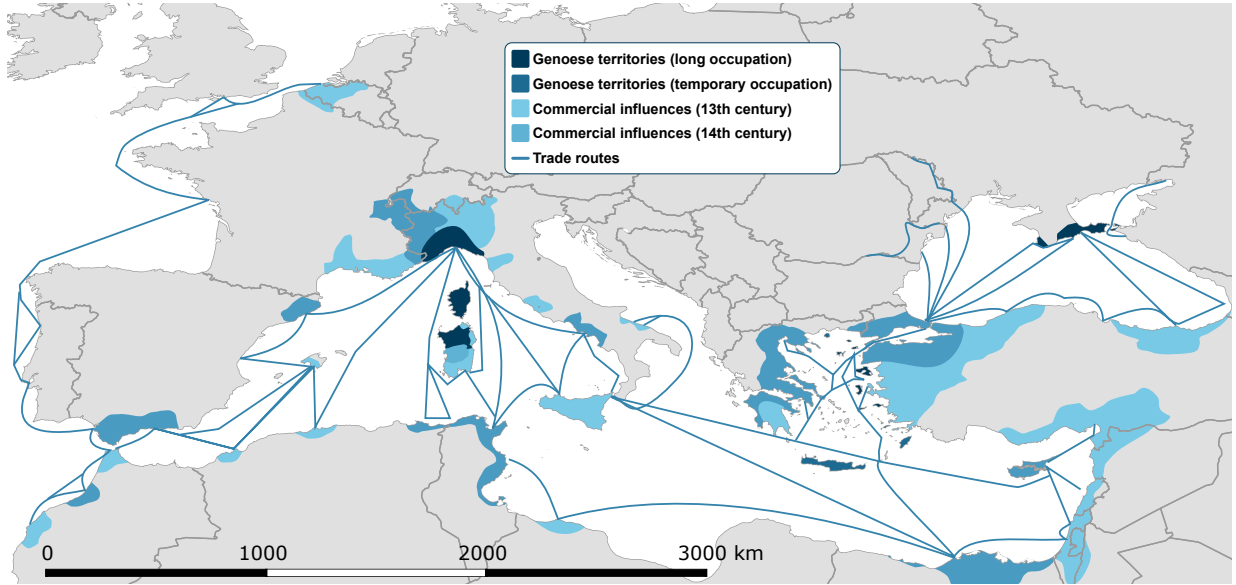


Figure 2.1: Expansion of Republic of Genoa around the Mediterranean and the Black Sea [66].

The maritime influence of the Republic of Genoa not only ensured the spread of the language across various Mediterranean territories but also facilitated significant lexical enrichment through contact with other languages. In particular, Genoese shares a notable amount of vocabulary with French, for instance, the word for ‘corkscrew’, *tirabuscion* in Genoese, clearly derives from the French *tire-bouchon*, standing in contrast to the Italian *cavatappi*. Additional significant influences can be traced back to Portuguese, as seen in the verb ‘to rain’, *ceuve* in Genoese and *chover* in Portuguese, which notably diverges from the Italian *piovere*. Similarly, Arabic impacts are evident in terms such as ‘longshoreman’, which in Genoese is *camallo*, derived from the Arabic *hammāl*, contrasting with the Italian *scaricatore*. These diverse lexical influences further contribute to a vocabulary that often diverges considerably from Standard Italian.

2.1.2 Main Linguistic Concepts

Term	Definition	Example
Diphthong	A sequence of two adjacent vowel sounds pronounced within the same syllable.	Italian <i>mai</i> (never); Genoese <i>figgeu</i> (boy)
Phoneme	The smallest unit of sound in a language that allows to distinguish one word from another.	Italian /n/ in <i>naso</i> (nose) and /v/ in <i>vaso</i> (vase); Genoese /c/ in <i>can</i> (dog) and /p/ in <i>pan</i> (bread)
Diacritics	Graphic marks added to letters to indicate changes in pronunciation, stress, or vowel length.	Accents (è, é, ê) and diaereses (ö)
Grave Accent	A diacritic (à, è, ì, ò, ù) used to indicate open vowel quality or primary stress.	Italian <i>caffè</i> (coffee); Genoese <i>pòrta</i> (door)
Acute Accent	A diacritic (á, é, í, ó, ú) used to indicate closed vowel quality.	Italian <i>perché</i> (why); Genoese <i>carréga</i> (chair)
Circumflex Accent	A diacritic (â, ê, î, ô, û) used to mark long vowels or historic sound contractions in Genoese.	Genoese <i>dormî</i> (to sleep)
Diaeresis	A diacritic (ä, ë, ï, ö, ü) used to mark a distinct vowel quality or to show that adjacent vowels are pronounced separately.	Genoese <i>öxello</i> (bird)
Standard Orthography	A formalized set of rules governing how a spoken language is consistently written, including spelling and diacritic conventions.	Standard Italian spelling rules versus the codified <i>grafia oficiâ</i> and <i>grafia unitäia</i> for Genoese
Clitic Pronouns	Unstressed pronouns that attach themselves to a verb. In Genoese, mandatory subject clitics have no direct equivalent in Italian.	Genoese <i>ti canti pròpio ben, ti!</i> (you sing really well!), where the first <i>ti</i> is the mandatory subject clitic differs from the Italian <i>canti proprio bene, tu!</i>

Table 2.1: Glossary of key linguistic, phonetic, and orthographic terms.

2.1.3 Orthography and Standardization Issues

A primary challenge in processing modern Genoese lies in the absence of a universally recognized official orthography. Due to its complex history of linguistic transformations and external influences, the language has not inherited a single, codified writing system.

Currently, several orthographic standards coexist, most notably the *grafia oficiâ*, proposed by the *Académia Ligùstica do Brénno* [1] and the *grafia unitäia*.

The latter was originally promoted by **Il Secolo XIX** [24], one of the most prominent Genoese media outlets and subsequently adopted by the ***Conseggio pe-o Patrimònio Linguistico Ligure*** (the Council for the Ligurian Linguistic Heritage) [12]. Alongside these formal standards, various informal conventions remain widely used by speakers. These systems differ significantly in their representation of phonemes, particularly in the use of diacritics, such as acute, grave and circumflex accents, alongside diaereses and in the transcription of diphthongs.

From a computational perspective, this lack of orthographic uniformity severely limits the availability of large, consistent corpora. In the context of **Neural Machine Translation (NMT)** [31, 28], this fragmentation introduces substantial noise during the training phase and complicates the evaluation process, as there is no single ‘ground truth’ for spelling, unlike in standardized languages such as Italian.

2.1.4 Morphosyntactic and Phonetic Divergences

One of the most significant grammatical differences between Genoese and Italian is the mandatory use of **subject clitic pronouns**. In Genoese, when a verb is conjugated in the second or third person singular, a subject clitic must precede the verb, even if an explicit subject pronoun is already present.

For instance, the second-person singular requires the clitic *ti* (or *t'* before vowels), as in the Genoese *ti ô veddi ti ascì?* for the Italian *lo vedi pure tu?* (Do you see it too?). Similarly, third-person singular verbs strictly mandate gender-specific clitics (*o* for masculine, *a* for feminine, with an euphonic *l'* before vowels), such as *o l'é anæto* (he went) or *a ven* (she comes). From a computational perspective, these structural divergences require the NMT model to master both the alignment of unaligned source-target tokens and the positional reordering triggered by verbal moods.

Beyond syntax, the languages differ significantly in their **phonetic-orthographic systems**. Genoese utilizes specific diacritics, such as the circumflex accent and the diaeresis, to represent vowel length and quality distinctions that do not exist in standard Italian. The presence of unique diphthongs and phonemes further separates the two languages. For a NMT system, these features increase the complexity of the **subword tokenization** process, as the model must accurately map the phonetic nuances of Genoese to maintain semantic correctness and orthographic consistency [12].

2.1.5 Data Scarcity and Digital Presence

Beyond morphosyntactic divergences and the lack of standardization, the primary obstacle to effective NMT is the status of Genoese as a **low-resource language**. In fact, Genoese (as part of the Ligurian language family) is classified by UNESCO as a **definitely endangered** language [41].

This status is primarily driven by sociolinguistic dynamics: as a language traditionally preserved through oral and familial transmission, its usage has continuously declined due to a weakening of intergenerational continuity. Without a strong tradition of formalized instruction or widespread written use, the pool of active native speakers has progressively contracted.

Consequently, Genoese suffers from a severely limited **digital footprint** and a total absence of large-scale, high-quality parallel digital corpora, resources that are readily available for high-resource language pairs like Italian-English. From an natural language processing perspective, training a neural model from scratch is hindered by this lack of data, preventing the network from reaching convergence. To mitigate this data scarcity, specialized strategies for low-resource settings become mandatory, such as **data augmentation** via **back-translation** to synthesize training pairs, and **transfer learning** through **fine-tuning** of pre-trained models on linguistically related language pairs.

2.2 Natural Language Processing

Natural Language Processing (NLP) is the branch of artificial intelligence dedicated to the interaction between computers and human languages. NLP encompasses a wide range of tasks, including **text classification**, **part-of-speech tagging**, **word segmentation**, **text-to-speech**, as well as the two primary tasks investigated in this thesis: **text normalization** and **machine translation**. For a comprehensive survey of foundational principles, task formulations, and historical methodologies across computational linguistics, we refer the interested reader to standard literature [28, 26].

Over the decades, the field has undergone a profound evolution: starting from early **rule-based (symbolic) systems**, it transitioned through **statistical machine learning** approaches before reaching the current dominance of **deep learning** architectures [52]. This shift has moved the focus from manually defined grammatical rules to the automatic extraction of complex linguistic patterns from large datasets using neural networks.

To process natural language computationally across any of these paradigms, text must first be transformed into continuous numerical representations suitable for machine learning models.

The remainder of this section reviews the fundamental methodologies governing data preparation and representation in modern NLP pipelines: specifically, we outline the evolution from sparse vector representations to dense contextual embeddings, examine various tokenization strategies, and explore back-translation as a primary data augmentation technique for machine translation tasks.

2.2.1 From One-Hot Encoding to Word Embeddings

A fundamental requirement for any machine learning model applied to natural language processing is the conversion of text into a numerical format, as neural networks cannot process raw strings of characters directly.

Historically, one of the primary methods for this conversion was **one-hot encoding** [32]. In this approach, each word in the vocabulary is represented as a high-dimensional sparse vector where only one element is set to 1 and the rest to 0. However, this method presents two significant limitations:

- **Dimensionality:** The size of the vector grows linearly with the vocabulary size, leading to enormous and computationally inefficient matrices, especially in large-scale corpora.
- **Lack of Semantics:** One-hot vectors are orthogonal by definition, meaning they cannot represent any relationship between words. Consequently, the model cannot perceive that words like ‘dog’ and ‘cat’ are semantically closer than ‘cat’ and ‘spoon’.

To address these shortcomings, **word embeddings** [32] were introduced. Unlike one-hot encoding, embeddings represent words as dense, **low-dimensional vectors** of real numbers. This technique is rooted in the **distributional hypothesis**, whose philosophical foundations date back to 1953, when the book *Philosophical Investigations* [68] by Ludwig Wittgenstein, which suggests that words appearing in similar contexts tend to share similar meanings, was published posthumously. In this continuous vector space, words with semantic affinities are positioned close to one another.

Early models such as **Word2Vec** and **GloVe** [40, 48] provided static representations, where each word had a single fixed vector. Modern architectures like **BERT** (Bidirectional Encoder Representations from Transformers) [16], however, have introduced **contextualized embeddings**. These allow the model to distinguish between different meanings of the same word based on its surrounding context, a feature that is essential for capturing the linguistic nuances of a complex, low-resource language like Genoese.

2.2.2 Tokenization Strategies

Tokenization is the process of decomposing a continuous stream of text into smaller, discrete units called **tokens**. These units can represent whole words, subwords, or individual characters. Tokenization serves as the essential preprocessing step that allows numerical models to construct meaningful embeddings from raw text.

The choice of tokenization level involves significant trade-offs:

- **Word-level Tokenization:** This approach splits text based on whitespace and punctuation. While it preserves the full semantic meaning of each token, it leads to excessively large vocabularies. Furthermore, it suffers from the **out-of-vocabulary** problem: if a word does not appear during the training phase, a common occurrence in low-resource languages like Genoese, the model is unable to process or translate it, often replacing it with a generic **unknown** token.
- **Character-level Tokenization:** By breaking text into individual letters or symbols, this method drastically reduces vocabulary size and eliminates the out-of-vocabulary problem, as any word can be represented as a sequence of known characters. However, individual characters lack inherent semantic meaning and the resulting sequences become significantly longer, making it harder for the model to learn long-range dependencies between words.
- **Subword Tokenization:** Currently the standard in modern natural language processing, this hybrid approach maintains frequent words as single tokens while decomposing rare or complex words into smaller sub-units. This ensures that common terms retain their semantic richness while unseen or rare words can be reconstructed from known sub-components.

A prominent technique in this category is **Byte Pair Encoding (BPE)** [55], that operates by iteratively identifying and merging the most frequent pairs of characters or character sequences within a corpus. For a morphologically rich and low-resource language like Genoese, Byte Pair Encoding is particularly effective: it allows the model to recognize shared roots and affixes across different word forms, effectively maximizing the utility of a limited dataset.

2.2.3 Back-translation and Data Augmentation

Beyond low-level sequence preprocessing techniques such as tokenization and vectorization, **back-translation** serves as a foundational data-centric strategy for **data augmentation** in NMT [54]. As previously discussed, the primary bottleneck for low-resource languages is the scarcity of parallel corpora. Since NMT architectures are highly data-intensive, back-translation provides a structured method to leverage abundant **monolingual data** to enhance translation quality.

The process follows a specific pipeline. Suppose the primary goal is to develop a translation system from a high-resource language (Source, e.g., Italian) to a low-resource language (Target, e.g., Genoese). The strategy involves three main steps:

1. **Inverse Model Training:** An initial, intermediate model is trained in the opposite direction (Genoese to Italian) using the limited parallel data available.
2. **Synthetic Data Generation:** This inverse model is then used to translate a large set of monolingual Genoese sentences into Italian. Even if the inverse model’s performance is modest, it produces a set of synthetic Italian-Genoese pairs.
3. **Augmented Training:** These synthetic pairs are combined with the original authentic parallel data to train the final target model (Italian to Genoese).

A crucial intuition behind this technique is the management of **noise**. In an Italian-to-Genoese task, providing the model with ‘synthetic’ Italian as input is often more effective than providing synthetic Genoese as the target. The decoder of the final model benefits from having authentic, high-quality target sentences (the original monolingual Genoese) to learn the correct language model of the target, while the neural architecture proves remarkably robust to the slight imperfections or ‘noise’ present in the synthetic source input.

2.3 Machine Learning for NLP

The application of machine learning to natural language processing is fundamentally rooted in the concept of **language modeling**. A language model is a mathematical framework designed to capture the statistical properties of human language.

The primary objective of a language model is predictive: given an observed sequence of tokens $(x^{(1)}, x^{(2)}, \dots, x^{(t)})$, the model aims to estimate the probability distribution of the subsequent token $x^{(t+1)}$, conditioned on the preceding context:

$$P(x^{(t+1)} | x^{(t)}, \dots, x^{(1)}) \tag{2.1}$$

By learning these conditional probabilities, the model can assign a likelihood to entire sequences of text, effectively modeling the underlying syntax and semantics of a language. In the current state-of-the-art, language modeling is predominantly addressed using deep learning architectures. Specifically, artificial neural networks allow for the representation of linguistic units in continuous vector spaces, enabling the model to capture complex, non-linear dependencies that were previously inaccessible to classical statistical methods.

2.3.1 Encoder-Decoder Architecture

The encoder-decoder architecture represents the standard framework for NMT, specifically designed to handle **sequence-to-sequence** tasks. This architecture consists of two primary components that work in tandem to map an input sequence of variable length to an output sequence of potentially different length.

The **Encoder** processes the input sequence from the source language (e.g., Italian). Its primary role is to compress the information of all input tokens into a high-dimensional representation known as the **context vector**. While word embeddings (such as those from Word2Vec) represent individual units, the context vector is a global representation of the entire source sentence, generated by the internal hidden states of the neural network.

The **decoder** utilizes this context vector to generate the output sequence in the target language (e.g., Genoese). The generation process is generally **autoregressive**, meaning the model predicts one token at a time, using both the context vector and the previously generated tokens as input for the next step. In the NMT framework, this architecture allows the model to transition from the abstract semantic representation of the source text to a grammatically and semantically correct sequence in the target language.

2.3.2 Recurrent Neural Networks and the Sequential Paradigm

The first successful implementations of the encoder-decoder framework relied heavily on **recurrent neural networks**, which, despite the architecture having been proposed earlier, was applied to NMT only in 2014. Unlike standard feed-forward networks, recurrent neural networks are designed to process sequential data by maintaining an internal **hidden state** that acts as a memory, updated at each time step as new tokens are consumed.

Among the various architectures, **long short-term memory** networks [21, 59] became the standard due to their ability to mitigate the **vanishing gradient problem** through a system of gates (input, forget, and output gates). These gates allow the model to regulate the flow of information, deciding which parts of the previous context should be preserved or discarded.

However, recurrent neural networks and long short-term memory networks suffer from a fundamental limitation: their **sequential nature**. Because each hidden state h_t depends on the previous state h_{t-1} , the computation cannot be parallelized, leading to long training times. Furthermore, as the distance between related tokens increases, the model struggles to maintain a precise representation of the earlier parts of the sequence, a phenomenon that directly contributes to the **bottleneck problem** addressed by the **attention mechanism**.

2.3.3 Attention Mechanism

The attention mechanism was introduced to overcome a major limitation of traditional sequence-to-sequence models [2], particularly those based on recurrent neural networks or long short-term memory networks. In standard architectures, the encoder is forced to compress the entire information of a source sentence into a single, fixed-length context vector. This creates a computational **bottleneck**: as the input sequence length increases, the vector loses the ability to preserve fine-grained semantic and positional details, leading to a significant degradation in translation quality.

The attention mechanism resolves this by allowing the decoder to access the full set of **hidden states** produced by the encoder during the processing of the input. This enables the model to selectively focus on the most relevant source tokens at each step of the generation process. The calculation of the attention score follows three fundamental stages:

1. **Alignment Score:** The model computes a similarity score between the current state of the decoder and each hidden state of the encoder, identifying which parts of the source sentence are most relevant to the token currently being generated.
2. **Attention Weights:** These scores are normalized using a **softmax** function. This ensures that the weights are positive and sum to one, effectively creating a probability distribution over the input tokens.
3. **Dynamic Context Vector:** A weighted average of the encoder's hidden states is calculated using the attention weights. This produces a dynamic context vector for each time step, providing the decoder with a tailored representation of the source text for every word it predicts.

The primary advantage of this approach is the improved handling of long-range dependencies. By establishing a direct and dynamic link between the source and target sequences, the attention mechanism ensures that the model maintains linguistic coherence regardless of sentence length, representing a crucial evolution in the field of NMT.

2.3.4 The Transformer Architecture

A significant milestone in the evolution of NMT was reached in 2017 with the transition from recurrent architectures to the **transformer** [63]. One of the inherent disadvantages of recurrent neural networks or long short-term memory networks is their sequential nature: tokens are processed one at a time, making it difficult to capture long-term dependencies. While bidirectional sequential networks attempted to mitigate this by processing sequences in both directions, they remained computationally inefficient and limited in their ability to model complex global contexts.

The transformer architecture fundamentally changes this approach by processing the entire input sequence simultaneously. This parallelization is made possible by the **self-attention** mechanism, which allows each token to attend to every other token in the sequence, regardless of their distance. Since the model lacks recurrence, it utilizes **positional encoding** to preserve the sequential order of the sentence. These vectors are added to the input embeddings to provide the model with information regarding the relative or absolute position of each token.

To fully characterize the transformer’s internal dynamics, it is necessary to detail the specific roles of the components within each encoder and decoder block:

- **Multi-Head Self-Attention:** Rather than computing a single attention representation, the model utilizes multiple ‘heads’ operating in parallel. Each head performs a linear projection of the **queries**, **keys** and **values**, enabling the model to simultaneously attend to information from diverse representation subspaces. For example, while one head may focus on short-range grammatical dependencies, another might capture broader semantic contexts. These independent outputs are then concatenated and projected back to the original dimensionality.
- **Position-wise Feed-Forward Networks:** Each transformer block includes a fully connected feed-forward network applied to each position independently. This component typically consists of two linear transformations separated by a non-linear activation function, such as **ReLU**. The primary function of the feed-forward network is to introduce non-linearity, allowing the model to learn more sophisticated patterns beyond the linear combinations provided by the attention mechanism.
- **Residual Connections and Layer Normalization:** To enable the effective training of deep networks, the architecture incorporates **residual connections** around each sub-layer. By defining the output as $\text{Sublayer}(x) + x$, the model can effectively mitigate the **vanishing gradient problem**, ensuring a more stable gradient flow during backpropagation. These connections are coupled with **layer normalization**, which standardizes the hidden states, preventing internal covariate shift and accelerating overall convergence.

Despite its superior performance, the primary challenge of transformer-based models is their intensive data requirement. For a low-resource language like Genoese, training a state-of-the-art model from scratch is often unfeasible due to data scarcity. Therefore, the most effective strategy is the adoption of **transfer learning** and **fine-tuning** [22]. By utilizing a model pre-trained on high-resource languages, specifically those sharing significant lexical and structural affinities with Genoese, such as Italian or Portuguese, it is possible to adapt the network to the target task. This approach effectively leverages pre-existing linguistic knowledge to overcome the limitations of a smaller dataset, exploiting the morphological similarities between these languages to facilitate the translation process.

2.3.5 Causal Language Modeling

In contrast to traditional bidirectional architectures or encoder-decoder sequence-to-sequence models, a **causal language model** [53] formulates text generation as an autoregressive next-token prediction task.

Given an observed sequence of tokens $X = (x_1, x_2, \dots, x_N)$, a causal model factorizes the joint probability of the sequence as the product of conditional probabilities using the chain rule [4]:

$$P(X) = \prod_{t=1}^N P(x_t \mid x_1, x_2, \dots, x_{t-1}) \quad (2.2)$$

During both training and inference, causality is strictly enforced by applying a **lower-triangular attention mask** (causal mask) [63] within the self-attention mechanism. This mask prevents token x_t from attending to any future tokens $x_{>t}$, ensuring that predictions at position t depend exclusively on the preceding context $(x_1, \dots, x_{t-1})$.

When applied to downstream tasks such as machine translation, causal **large language models** [5] process the task without explicit cross-attention mechanisms. Instead, the source sentence S and target sentence T are concatenated into a single input sequence, forcing the network to autoregressively generate the target sequence conditioned causally on the source context.

2.3.6 Large Language Models

In recent years, the scaling of transformer architectures, combined with massive amounts of pre-training data and computational capacity, has given rise to large language models. Unlike traditional task-specific encoder-decoder models, large language models are typically built on decoder-only transformer architectures, such as the **GPT** series [5], **LLaMA** [62], or **Mistral** [25], trained autoregressively on next-token prediction over heterogeneous, multi-billion-token corpora.

Through this large-scale generative pre-training, large language models acquire broad world knowledge and strong contextual reasoning capabilities, allowing them to perform a wide array of natural language processing tasks, including translation, summarization, and text normalization, via zero-shot or few-shot **prompting**, without necessitating architectural task-specific modifications. Furthermore, techniques such as **instruction tuning** [64] and **reinforcement learning from human feedback** [46] align these models to follow complex natural language instructions and adapt to downstream requirements.

Despite their state-of-the-art performance across high-resource domains, applying large language models to regional and **low-resource languages** like Genoese introduces distinct challenges:

- **Tokenization Disparity and Subword Fragmentation:** Tokenizers of general-purpose LLMs are predominantly optimized for high-resource languages (e.g., English, Spanish, Italian). When processing low-resource or dialectal text with unique orthographic conventions or diacritics, the tokenizer splits words into excessively small subwords or individual characters. This severely increases the sequence length, reduces context efficiency, and elevates computational overhead.
- **Hallucination and Over-Standardization:** Because large language models are heavily exposed to dominant languages during pre-training, they exhibit a bias toward standard forms (such as standard Italian). In tasks involving regional normalization or translation, large language models may accidentally over-standardize native phonetic structures or hallucinate plausible yet unauthentic vocabulary.
- **Fine-Tuning Costs vs. Specialized Architectures:** While **parameter-efficient fine-tuning** methods, such as **low-rank adaptation** [23], allow large language models to be adapted to specific targets, training or fine-tuning models with billions of parameters remains computationally expensive compared to compact, specialized sequence-labeling or sequence-to-sequence architectures.

In the context of non-standard text processing and regional language normalization, evaluating whether generalist Large Language Models can match or surpass lightweight, task-dedicated architectures remains a critical research question.

Chapter 3

Dataset

Data represent a crucial component of the entire experimental pipeline of this thesis. To address our methodological objectives, we collected and integrated multiple heterogeneous data sources spanning different historical periods and presenting diverse formats. In this chapter, we describe these sources and their primary features.

3.1 Data Sources

We partially carried out the data collection phase and the initial aggregation of the sources in collaboration with Sara Caviglia and Fabrizio Sardo [9, 17] as part of a project developed for the Natural Language Processing course.

To capture the full spectrum of the Genoese language, we categorized the collected resources into four distinct domains: online dictionaries, web media, preexisting parallel corpora, and historical heritage texts.

3.1.1 Online Dictionaries

- ***Diçionäio eletrònico italian-zeneise (DEIZE)***: The official Italian-Genoese online dictionary developed by the ***Conseggio pe-o Patrimònio Linguistico Ligure*** [13, 12]. In addition to an extensive lemma list of parallel Italian-Genoese words, this resource includes a vast collection of contextual examples written according to the standardized *grafia unitäia*.

- **Traduttore Italiano Genovese (TIG):** An online dictionary curated by Franco Bampi [3], adhering to the orthographic criteria of the *grafia oficiâ*. It consists of aligned Italian-Genoese lemma pairs.

3.1.2 Web Corpora and Digital Outlets

- **O Zinâ:** An online weekly journal published by the *Conseggio pe-o Patrimònio Linguistico Ligure* [45, 12] in *grafia unitäia*, featuring news articles in Genoese alongside their Italian translations.
- **Lunäio Zeneise da sciâ Tägnâ:** A cultural blog dedicated to Ligurian traditions and local folklore [33]. It consists mainly of monolingual texts, though it also contains several bilingual sections with parallel Italian translations.
- **Wikivivagna:** The Ligurian-language edition of **Wikisource** [67], hosting a digital web corpus of prose and poetry in Genoese characterized by high orthographic and temporal variability.

3.1.3 Preexisting Parallel Corpora

- **ZenaMT:** A pre-aligned parallel Italian-Genoese corpus compiled for previous research by Haberland et al. [20], aggregating various digital and translated texts. Created in collaboration with the *Conseggio pe-o Patrimònio Linguistico Ligure*, the dataset primarily consists of sentence pairs sourced from *O Zinâ*.

3.1.4 Historical and Heritage Texts

This category comprises 19th and 20th-century heritage texts that present a high degree of linguistic complexity. For each source, prior research by Lusito et al. [36] provided both the original historical version and a normalized transposition into modern *grafia unitäia*, curated by expert linguists:

- **Casaccia:** Examples extracted from the 19th-century *Dizionario Genovese-Italiano* by Giovanni Casaccia [8].
- **Piaggio:** The 19th-century poetic compositions in verse by Martin Piaggio [49].
- **O Balilla:** Archival excerpts from a prominent Ligurian bi-weekly newspaper published between the late 19th and early 20th centuries [44].

- **Gazzo:** Five cantos extracted from Angelico Federico Gazzo’s historical Genoese translation of Dante Alighieri’s *Divine Comedy* [19].

3.2 Source Parsing and Format Preprocessing

A significant challenge in our data collection phase arose from the structural heterogeneity of the raw sources. We acquired data in disparate file formats depending on their origin: unstructured web-scraped **HTML**, plain **TXT** files, compressed **Parquet** tables, aligned **.raw/.norm** text pairs, and standard **CSV** spreadsheets. To resolve these inconsistencies, we developed and deployed source-specific parsing and cleansing sub-pipelines to convert each format into a unified representation.

3.2.1 Plain Text Parallel Alignment (*O Zinâ* and *Lunäio*)

We processed the resources from *O Zinâ* and *Lunäio Zeneise da sciâ Tägnâ* [45, 33], which were originally provided as separate plain text (**TXT**) files for each language layer.

- **Sentence-Pair Reconstruction:** We reconstructed parallel structures via a deterministic line-by-line parsing routine, mapping each Italian sentence to its exact positional counterpart in the Genoese file.
- **Cleansing and Deduplication:** We programmatically removed empty lines and padding whitespace, applying basic standardization steps such as lowercasing and exact-match deduplication to eliminate redundant pairs.

3.2.2 Columnar Data Decompression (ZenaMT)

We processed the ZenaMT corpus [20], which we acquired in **Parquet** format, a compressed columnar storage layout optimized for large-scale processing frameworks.

- **Extraction and Mapping:** We decompressed the files and converted them into a flat tabular format, directly mapping preexisting translation pairs to our target schema fields.
- **Standardization:** We routed the extracted rows through the same cleansing routines applied to our text-based parallel corpora.

3.2.3 CSV Columnar Extraction (DEIZE)

We extracted data from the *DEIZE* resource [13], which contains both lemmatized word pairs and contextual sentence-level examples natively provided in **CSV** format.

- **Feature Selection:** We implemented a targeted script to isolate the core translation columns (**Italian** and **Genoese**) while systematically stripping supplementary metadata, descriptive annotations, and original spreadsheet indexing.
- **Filtering:** We cleansed the extracted text pairs by stripping padding whitespace, lowercasing, and deduplicating entries.

3.2.4 HTML DOM Parsing and Polysemic Expansion (TIG)

We scraped the *TIG* online dictionary [3] in raw **HTML** format consisting of nested web layouts.

- **DOM Parsing and Link Filtering:** We parsed Document Object Models using regular expressions and tag-filtering libraries to isolate lemma containers. We systematically discarded cross-referencing anchor tags (**<a>**) linking to synonymous words without explicit translations.
- **Polysemic Expansion:** When a single Italian lemma mapped to multiple Genoese translations within the same HTML block, we applied a flattening routine to duplicate the Italian source lemma across separate rows. This preserved the full polysemic and synonymic diversity within a flat structure.
- **Cleansing:** We stripped isolated pairs of residual HTML entities, lowercasing and deduplicating the final results.

3.2.5 Monolingual Sentence Extraction (Wikivivagna)

We processed the *Wikivivagna* corpus [67], which was provided as unstructured plain text (**TXT**) files containing raw Genoese prose and poetry.

- **Sentence Segmentation:** We implemented a text segmentation routine to split the continuous stream of text into individual periods and sentences.
- **Monolingual Schema Population:** Since no target translations were present, we ingested the extracted sentences into the master file, leaving the **Italian** field blank and setting the **Type** flag to **Monolingual**.

3.2.6 Historical Layer Mapping (Casaccia, Piaggio, O Balilla, and Gazzo)

We integrated the historical heritage corpora (*Casaccia*, *Piaggio*, *O Balilla*, and *Gazzo*) [8, 49, 44, 19], provided as aligned file pairs in **.raw** (original historical orthography) and **.norm** (normalized transposition into modern *grafia unitäia*) formats.

- **Paired File Parsing:** We developed a custom parser to ingest corresponding **.raw** and **.norm** files synchronously, matching each historical sentence snippet with its standardized counterpart.
- **Schema Integration:** We directly wrote the aligned pairs to the master **CSV** file, preserving both the original variant and its modern normalized representation for our training sub-pipelines.

3.3 Master Dataset Schema and Aggregation

Following format-specific conversion, cleansing, and exact-match deduplication routines, we aggregated all processed resources into a single relational master file.

3.3.1 Relational Schema

Each record in the consolidated dataset represents a distinct textual unit (word pair or sentence) governed by seven explicit features:

- **Id:** Unique numeric identifier ensuring cross-pipeline traceability.
- **Italian:** Standard Italian text snippet (left blank for monolingual Genoese records).
- **Genoese:** Genoese text snippet preserving its original input orthography.
- **Source:** Categorical string indicating data provenance.
- **Level:** Granularity indicator distinguishing word-level (**word**) from sentence-level (**sentence**) tokens.
- **Type:** Classification flag set to either **Bilingual** or **Monolingual**.
- **Grafia:** Label tracking the specific orthographic convention or historical layer.

3.3.2 Corpus Statistics and Structure

The final consolidated master dataset comprises a total of 265886 records. An illustrative subset demonstrating the unification of diverse sources under this schema is presented in Table 3.1.

Id	Italian	Genoese	Source	Level	Type	Grafia
31480	rifare	refâ	TIG	word	Bilingual	Ofiçiâ
59192	porzione di focaccia	slerfa de fugassa	DEIZE	word	Bilingual	Unitäia
76781	il genoa ha segnato due reti nel primo tempo	o zena o l'â fæto doe ræ into primmo tempo	DEIZE	sentence	Bilingual	Unitäia
80801	la madre ha dichiarato: «ora è un angelo ma voglio giustizia, lo ricorderò con una fondazione»	a moæ a l' â dito: «oua o l'é un angeo ma veuggio giustizia, l'arregordiò con unna fondaçion»	O Zinâ	sentence	Bilingual	Unitäia
241829		no se queta, ciû ûn momento	Wikivivagna	sentence	Monolingual	Other

Table 3.1: Example records from the consolidated master dataset schema.

The label **Other** in the **Grafia** feature accounts for entries that lack a specific orthographic designation or diverge from both the standardized *grafia unitäia* and *grafia ofiçiâ* conventions.

To provide a granular quantitative breakdown of the consolidated corpus, Figure 3.1 illustrates the structural distribution of records across four key categorical dimensions: orthographic standard (**Grafia**), textual granularity (**Level**), linguistic pair type (**Type**), and origin corpus (**Source**).

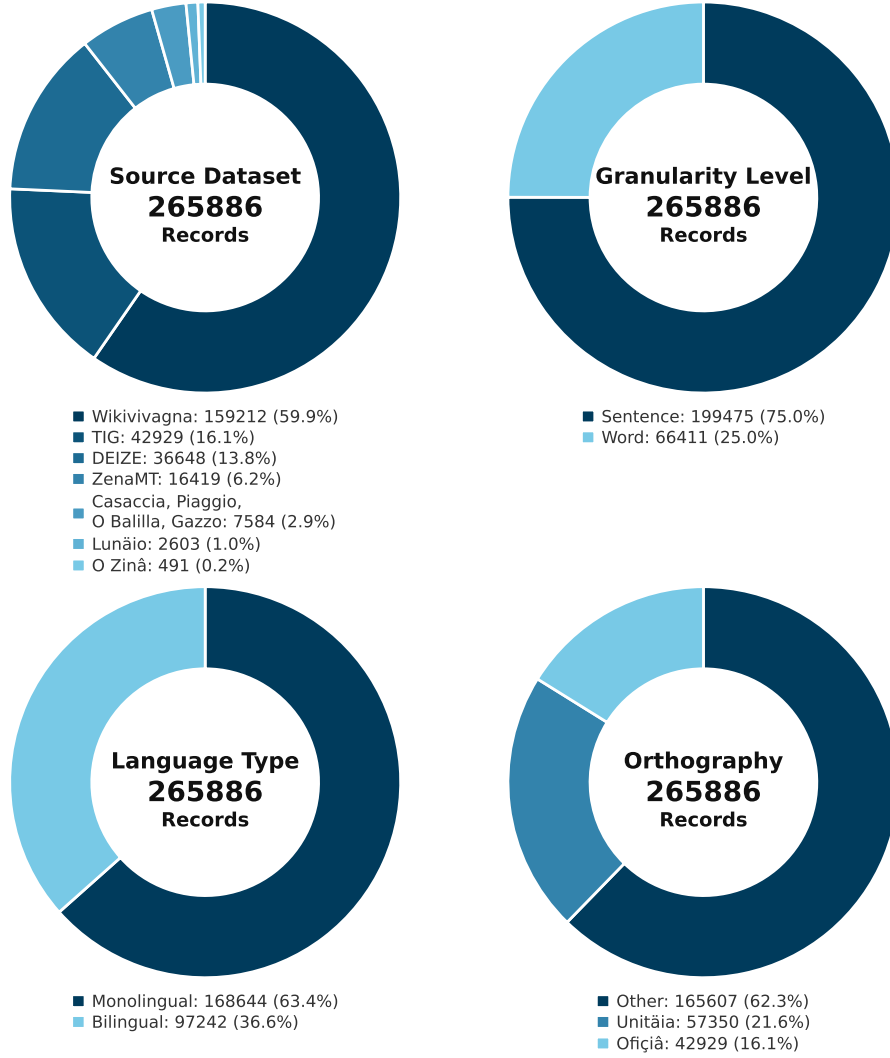


Figure 3.1: Proportional breakdown of the master dataset across key categorical dimensions (Grafia, Level, Type, and Source).

As shown in the subplot analysis of Figure 3.1:

- **Source:** The dataset aggregates records across diverse provenance nodes, dominated by continuous text from *Wikivivagna* (159,212 records, 59.9%), combined with lexicographic sources and pre-existing corpora such as *TIG* (42,929 records, 16.1%), *DEIZE* (36,648 records, 13.8%), *ZenaMT* (16,419 records, 6.2%), historical literary works (*Casaccia*, *Piaggio*, *O Balilla*, *Gazzo*, 7,584 records, 2.9%), *Lunăio* (2,603 records, 1.0%), and *O Zină* (491 records, 0.2%).

- **Level:** The distribution establishes a clear structural balance, with three-quarters of the corpus comprising continuous contextual text at the **Sentence** level (199475 records, 75.0%) and the remaining quarter consisting of isolated lexical items at the **Word** level (66411 records, 25.0%).
- **Type:** The corpus incorporates a majority of **Monolingual** Genoese sequences (168644 records, 63.4%) primarily dedicated to target-side language modeling, alongside a substantial portion of **Bilingual** parallel pairs (97242 records, 36.6%) leveraged for cross-lingual tasks.
- **Grafia:** The orthographic breakdown shows that non-standardized variants labeled as *other* represent the largest share (165607 records, 62.3%), while standardized spelling conventions account for the remaining portion, divided between *grafia unităia* (57350 records, 21.6%) and *grafia oficiă* (42929 records, 16.1%).

This centralized structure serves as the foundation for all subsequent processing stages, enabling dynamic filtering and partitioning to generate specialized sub-datasets for both normalization and translation tasks.

3.4 Dataset Partitioning and Task Alignment

To generate optimal, task-specific training sets while maintaining full traceability to the master schema, the dataset is dynamically partitioned into two target sub-datasets:

- **Text Normalization Task:** Extracts entries across diverse orthographic conventions to train models that map non-standard Genoese variants into the standardized modern format.
- **Machine Translation Task:** Filters high-purity bilingual pairs alongside selected monolingual Genoese sentences (for back-translation and pre-training) to build the dataset for neural machine translation models.

Further details regarding the specific datasets utilized for the normalization and translation tasks will be provided in Chapters 4 and 5, respectively, following a more detailed definition of each task and its specific requirements.

Chapter 4

Normalization

Text normalization is the computational process of transforming non-standard, highly variable textual inputs into a single, standardized orthographic representation. This step is particularly vital for regional and low-resource languages such as Genoese, which lack a single, universally enforced orthographic convention. Throughout history, written Genoese has undergone continuous mutations, heavily influenced by individual choices, phonetic transcriptions, and local dialectal traditions of various authors and translators.

Such high lexical and graphemic variability significantly complicates NLP tasks, ranging from part-of-speech tagging and text-to-speech synthesis to neural machine translation. When a single word can be instantiated in multiple distinct surface forms, downstream models face severe vocabulary fragmentation and artificial data sparsity. Furthermore, this fragmentation compromises automated evaluation metrics, as reference corpora cannot practically account for every historical or regional spelling variant. Transforming raw dialectal text into a unified baseline is therefore a necessary prerequisite for robust computational modeling and accurate downstream evaluation.

4.1 Related Works in Text Normalization

This section reviews the evolution of text normalization methodologies, tracing the shift from traditional statistical and rule-based pipelines to modern character-level sequence-to-sequence neural architectures, with a specific focus on dialectal normalization and Genoese orthographic standardization.

4.1.1 Evolution of Text Normalization

Early non-neural approaches framed text normalization primarily as an unsupervised mapping task to convert informal or non-standard tokens into standardized canonical forms. For instance, Yang and Eisenstein [71] introduced an unsupervised log-linear model paired with a target language model, using a Sequential Monte Carlo sampling algorithm guided by orthographic distance features to learn noise patterns without annotated parallel data.

Rule-Based, Subword, and Multi-Stage Pipelines

With the emergence of neural sequence-to-sequence architectures and weighted finite-state transducers, early frameworks cast normalization into specialized translation or string transduction tasks. Sproat and Jaitly [58] addressed text normalization for speech synthesis by processing character sequences within sliding context windows using bidirectional LSTMs, constraining the model’s lattice with finite-state transducers to mitigate catastrophic hallucinations. In low-resource scenarios characterized by noisy web-scraped data and inconsistent orthographic standards, finite-state grammar frameworks have proven effective, for instance, Zupon et al. [75] applied language-specific rewrite rules to normalize a diverse suite of low-resource African languages, effectively reducing dataset perplexity and filtering orthographic noise. To balance character-level granularity with full sequence context, subword tokenization strategies were also introduced. Mansfield et al. [37] formulated text normalization as a neural machine translation task operating on subword units generated via byte pair encoding [55], mitigating out-of-vocabulary explosion while embedding positional and part-of-speech features. Concurrently, hybrid multi-stage frameworks (e.g., Yolchuyeva et al. [73]) combined deep learning entity classifiers (BiLSTMs and convolutional neural networks) with deterministic rule-based verbalizers to enforce structural constraints on complex semiotic classes.

Character-Level Dialectal Transduction

Text normalization operates primarily as a **character-level sequence-to-sequence transduction task**. Because non-standard regional variants and their normalized counterparts share underlying lexical roots and morphosyntactic structures, the mapping relies heavily on fine-grained graphemic transformations rather than arbitrary lexical substitution.

Operating directly on raw character sequences eliminates out-of-vocabulary bottlenecks caused by unpredictable dialectal spelling variants. Bidirectional LSTM networks augmented with dynamic attention mechanisms [2] have established a core benchmark for character-level transductions.

By encoding contextual character representations bidirectionally and dynamically aligning source and target graphemes, BiLSTM architectures effectively capture complex morphological, phonetic, and accentual mappings across non-standard orthographic systems.

In the context of dialectal normalization and historical manuscripts, where annotated training data is severely limited, character-level models have proven highly effective. Lusetti et al. [35] addressed the normalization of informal Swiss German WhatsApp messages into standard High German using a character-level sequence-to-sequence architecture with bidirectional LSTMs and soft attention, integrated with a secondary word-level model to verify word-boundary validity.

4.1.2 Genoese Orthographic Normalization

Computational resources and processing pipelines specifically targeting Ligurian and Genoese have historically been constrained by dialectal fragmentation and orthographic instability. Recent empirical evaluations by Buscaldi and Rosso [7] on Genoese corpora demonstrated how graphemic variance and vocabulary mismatch severely limit the direct zero-shot transfer of multilingual sequence-to-sequence architectures.

Addressing Genoese orthographic standardization, Lusito, Maillard, and Ferrante [36] established a crucial state-of-the-art benchmark. Their work focused on normalizing 19th-century Genoese literary texts into the modern standardized orthography (*grafia unitäia*). Utilizing transformer-based NMT architectures operating at the subword level, their models mapped archaic and non-standard spelling variations into normative conventions, demonstrating the feasibility of neural transduction for digital preservation. Building upon these foundations, our work extends Genoese text normalization by formulating the task directly at the character level, benchmarking probabilistic sequence labelers against attention-augmented BiLSTM architectures across both isolated lexical items and full contextual sentences.

4.2 Data Processing for Text Normalization

The primary objective of the text normalization step is to unify heterogeneous orthographic variants into a single, standardized writing system. The master dataset comprises three distinct orthographic conventions: *ofiçiâ*, *unitäia*, and *other*. The *other* category, as briefly mentioned in Section 3.3.2, aggregates non-standardized orthographies that lack formal codification or diverge from official standards, including historical, popular, and informal variants. Consequently, the normalization pipeline could target either the *ofiçiâ* or the *unitäia* convention.

As shown in Table 4.1, *unitäia* represents the predominant codified standard within the dataset and was therefore selected as the **target orthography**.

Orthographic System	Number of Entries
<i>Other</i>	165607
<i>Unitäia</i>	57350
<i>Ofiçiâ</i>	42929

Table 4.1: Distribution of entries across writing systems in the master dataset.

When the thesis project started, aligned sentence pairs from the historical sources (*Casaccia*, *Piaggio*, *O Balilla*, and *Gazzo*) [8, 49, 44, 19] were unavailable. Consequently, the normalization pipeline was at first directed toward word-level alignment rather than full sentence normalization. We thus constructed a dedicated parallel dataset from the primary dataset, described in Chapter 3, by aligning individual lexical entries from *DEIZE*, *TIG*, and *Wikivivagna* [13, 3, 67].

4.2.1 Normalization Dataset

To train and evaluate the character-level sequence-to-sequence normalization models, two distinct parallel corpora were constructed across different levels of granular resolution: a **word-level dataset** and a **sentence-level dataset**. The word-level dataset relies on automated heuristic alignment pipelines across lexicographical and unstructured sources, whereas the sentence-level dataset incorporates manually validated historical parallel texts.

Word-Level Alignment via Dictionary Merging

To align the *TIG* (*grafia ofiçiâ*) and *DEIZE* (*grafia unitäia*) Genoese-Italian dictionaries into a coherent parallel word-level dataset, a multi-step pipeline was executed in chronological order:

1. **Exact Lemma Matching and Slash Resolution:** An initial relational join mapped identical Italian source entries across both dictionaries. Immediately following this merge, entries containing alternative lexical variants separated by slashes (e.g., *option1/option2*) were systematically resolved. Each variant was evaluated individually against the opposing dictionary entry, retaining only the single option that maximized structural proximity while discarding remaining candidates.

2. **Normalized Distance Calculation and Threshold Filtering:** A normalized Levenshtein distance was computed for candidate pairs by dividing the edit distance by the maximum token length. Strings underwent Unicode normalization to strip diacritics and hyphens, ensuring comparison focused purely on graphemic sequences. Pairs exceeding a strict normalized distance threshold of 0.3 were discarded to eliminate noisy alignments.
3. **Cross-Dictionary Disambiguation:** Duplicate mappings for Genoese lemmas across dictionaries were resolved by sorting remaining candidates in ascending order of edit distance. The pipeline systematically dropped lower-ranked duplicates, preserving exclusively unique translation pairs that maximized lexical similarity.

Word-Level Alignment via Vocabulary Sifting

To expand the word-level normalization dataset using unstructured sources, a dedicated alignment pipeline mapped localized Genoese vocabulary from *Wikivivagna* sentences against the consolidated dictionary lemmas through the following workflow:

1. **Sentence Tokenization and Text Cleaning:** Full sentences extracted from *Wikivivagna* were segmented into word tokens. Boundary punctuation marks (e.g., quotation marks, periods, commas) were programmatically stripped while preserving internal apostrophes and hyphens essential to Genoese orthography. Duplicate tokens were removed to form a unique target vocabulary.
2. **Token-Count Grouping and Structural Pre-Filtering:** Dictionary lemmas and *Wikivivagna* tokens were segmented and grouped by internal sub-token counts (splitting on spaces and hyphens). Pairwise matching was restricted strictly to strings sharing identical sub-token counts, drastically reducing search complexity.
3. **Vectorized Distance Matrix Computation:** Within each structural group, a parallelized distance matrix was computed using low-level optimized bindings. Normalized Levenshtein distances were calculated on diacritic-insensitive representations alongside an absolute length delta matrix.
4. **Heuristic Sifting and Candidate Reduction:** Alignment candidates were tentatively retained only if they satisfied a dual-gate constraint: a normalized Levenshtein distance strictly below 0.3 and an absolute length difference not exceeding 3 characters.

5. **Morphological Alignment and Quality Thresholding:** Verification enforced that initial and final base graphemes (independent of diacritics) matched perfectly between paired elements. Finally, the acceptance threshold for the normalized Levenshtein distance was tightened to a strict maximum of 0.1. Without an Italian reference word, higher distance thresholds allowed false alignments between distinct lexical items with high surface similarity, such as *casa* (home) and *cosa* (thing, whose normalized form *cōsa*) yields a normalized Levenshtein distance of 0.25.

Sentence-Level Parallel Dataset

The sentence-level component of the normalization dataset comprises full parallel sentences extracted from the historical heritage collection (*Casaccia*, *Piaggio*, *O Balilla*, and *Gazzo*). These resources provide direct sentence-aligned mappings between original historical orthographic variants (*other*) and their expert manual transliterations into standardized modern *grafia unitäia*.

4.2.2 Dataset Splitting

To evaluate the generalization capacity and robustness of the proposed normalization frameworks, the datasets were partitioned into distinct subsets tailored to the structural constraints of the task.

Data splitting strategies were applied across two primary granularities derived from the global corpus: the word-level dataset, consisting exclusively of standalone lemmas, and the sentence-level dataset, both detailed in Section 4.2.1, comprising full contextual sequences.

To guarantee exact experimental reproducibility across all approaches, a deterministic partitioning strategy was enforced using a fixed random seed. For both word-level and sentence-level corpora, alignment pairs were divided according to a standard 80%/10%/10% ratio for Training, Validation, and Test sets, respectively. The final quantitative breakdown is summarized in Table 4.2.

Dataset	Total Records	Train	Validation	Test
Word-Level	18228	14597	1814	1817
Sentence-Level	3791	3030	380	381

Table 4.2: Quantitative distribution of baseline records across the experimental data partitions for the normalization task.

4.3 Methods

Normalization is a crucial step in unifying the writing systems of a low-resource language. Since Genoese lacks millions of parallel sentences required to train a translation model entirely from scratch, it is imperative to leverage all available data. However, exposing a model to a high degree of orthographic variation in its inputs significantly hinders its capacity to learn accurate word translations. Consequently, the primary objective is to align all available textual data into a single, standardized writing system.

To determine the target convention, non-standardized options must first be ruled out. Converting texts into the *other* category is mathematically and linguistically unfeasible, as it does not represent a cohesive writing system governed by formal rules, but rather acts as a generic container for heterogeneous orthographic variations. Concurrently, as demonstrated in Table 4.3, the *ofiçiâ* orthography contains zero records at the sentence level in the global dataset.

Orthographic System	Number of Entries
<i>Other</i>	161815
<i>Unitäia</i>	30076
<i>Ofiçiâ</i>	0

Table 4.3: Number of entries at the sentence level per each writing system.

This complete absence of sentence-level data for *ofiçiâ*, combined with the structural fragmentation of the *other* category, leaves the *grafia unitäia* as the sole viable target system. Beyond being the primary codified standard in the lexicon (Table 4.1), *unitäia* accounts for all available standardized sentence-level entries (Table 4.3). Consequently, the ultimate goal of the normalization framework is to systematically transduce sentences belonging to the *other* classification into the standardized, rule-governed framework of the *grafia unitäia*.

4.3.1 Task Formulation

To reach this objective, the orthographic normalization process is formalized as a character-level sequence-to-sequence transduction task. To systematically evaluate the generalization capabilities of the architecture, the task was investigated and formulated at two distinct levels of linguistic granularity:

1. **Word-Level Normalization:** The framework is initially applied to isolated word tokens. Given an input sequence of characters representing a single unstandardized word $W_{in} = (x_1, x_2, \dots, x_n)$ from the *other* or *grafia oficiã* vocabularies, the model learns a mapping function $f : W_{in} \rightarrow W_{out}$ to generate the corresponding standardized target word $W_{out} = (y_1, y_2, \dots, y_m)$ in *grafia unitãia*. This granular formulation isolates the morphological and phonetic transformations, focusing strictly on localized graphemic shifts without context variables.
2. **Sentence-Level Normalization:** The paradigm is subsequently scaled to full textual sequences to handle realistic translation inputs. Given an entire sentence sequence $S_{in} = (x_1, x_2, \dots, x_n)$, the objective is to generate the optimal standardized target sentence $S_{out} = (y_1, y_2, \dots, y_m)$. This advanced formulation requires the model to not only map character transformations but also maintain contextual consistency across syntactic boundaries, handling word spacing and punctuation fluidly.

Operating explicitly at the character level across both granularities, rather than at the traditional word or sub-word token level, is a fundamental architectural constraint. In low-resource scenarios, word-level vocabularies suffer from severe data sparsity and high out-of-vocabulary rates. More importantly, while machine translation involves mapping entirely different lexical items and syntactic shifts across languages, orthographic normalization deals with variations where the underlying word and its semantic meaning remain constant and only specific character configurations change. By compressing the structural space to a closed vocabulary $\mathcal{V}$ consisting solely of localized graphemes, digits, and diacritics, the framework shifts the objective from a complex semantic translation problem to a specialized phonetic and orthographic transposition task. This allows the model to fluidly generalize over unseen lexical roots by learning recurring character substitutions, insertions and deletions, whether processing an isolated lemma or a complex syntactical sentence.

To address the character-level transduction task, three distinct computational methods were investigated: the first two focused exclusively on the word level, while the third encompassed both.

4.3.2 Norm-RB: Deterministic Rule-Based Extraction

The first method relies on a deterministic, rule-based framework designed to automatically extract and apply character-level transformation patterns from parallel corpora. The pipeline is organized into four core stages:

1. **Character Alignment (Needleman-Wunsch):** To map the exact transitions from a non-standard source string (S) to its official target (T), the system employs the Needleman-Wunsch global alignment algorithm [43, 11]. Since this alignment serves as a deterministic preprocessing step rather than a core hyperparameter of the normalization task, it is parameterized using a standard affine gap penalty configuration. Specifically, the scoring matrix is set to +3.0 for a character match and -1.5 for a mismatch. To prevent excessive and fragmented character splits, the gap scoring model penalizes the introduction of new gaps with an opening score of -4.0, while the continuation of an existing gap is penalized with an extension score of -1.0. In cases where the algorithm yields multiple optimal paths with identical maximum scores, a right-leaning tie-breaking heuristic is applied, forcing gap characters (-) toward the end of the string to ensure geometric consistency across the extracted patterns.
2. **Contextual Rule Modeling:** Formally, each structural modification is abstracted and uniquely identified as an operation tuple:

$$R = \langle op, src, dst, left, right \rangle$$

where $op \in \{\text{SUB}, \text{INS}, \text{DEL}\}$ denotes the operation type (Substitution, Insertion, or Deletion), src and dst represent the source and destination substrings (which may be empty for insertion and deletion operations, respectively) and $left$ and $right$ capture the immediate context within a dynamic window of size w . Boundary effects at the word edges are handled via virtual padding using anchor symbols (^ for word-start and \$ for word-end). While the implementation carries additional metadata for tracking and provenance purposes, such as sample words and length attributes, the mathematical identity of a rule is strictly bounded by this quintuple. To maintain strict contextual integrity during multi-step extractions within the same word, the source string is updated incrementally, ensuring that subsequent context evaluations reflect localized changes already identified.

3. **Statistical Scoring and Priority Assignment:** For each unique rule harvested from the training dataset, the extraction pipeline aggregates the localized instances into a comprehensive knowledge base. To evaluate the empirical weight and reliability of each rule, the system computes an array of statistical metrics:

- **Frequency** (f): The absolute number of times the specific transformation is observed across the entire dataset.
- **Lexical Spread** (n_{words}): The number of distinct unique words in which the rule appears, serving as an indicator of whether a pattern is systemic or confined to specific lexical items.
- **Average Word Length** (l_{avg}): The mean length of the source words that triggered the rule, providing morphological context regarding the token structure.
- **Confidence** (c): The conditional probability of the rule’s application given its specific source context (defined by the combination of *src*, *left* and *right* for a given window size w). It represents the local determinism of the shift and is calculated as:

$$c = \frac{f}{\text{Total occurrences of the matching source context}}$$

During the inference phase, rules are ordered and selected via a composite **Priority** score (P). This score is formulated as a weighted linear combination that maps the specificity of the context (window size w), the local reliability (confidence c) and a logarithmically scaled factor of the empirical volume (absolute frequency f). Additionally, an operational bias factor ω_{op} is introduced to modulate priority based on the operation type, allowing the system to penalize specific structural shifts (such as insertions or deletions) relative to simple substitutions to mitigate uncontrolled string-length drift.

4. **Right-to-Left Inference:** When normalizing an unseen word, the system applies the prioritized rule queue using a greedy strategy. Crucially, transformations are executed in a *right-to-left* direction. This positional constraint ensures that any insertions or deletions do not invalidate or shift the index offsets of pending character sites further down the string. The inference loop runs iteratively until a fixed point is reached (no further rules match) or a maximum safety threshold of operations is exceeded to prevent infinite cycles.

4.3.3 Norm-CRF: Conditional Random Fields

To transition from rigid heuristics to probabilistic machine learning, the second method formalizes word-level normalization as a structured sequence labeling task using Conditional Random Fields (CRF). By modeling the conditional distribution of character-level tags, this approach attempts to capture statistical transitions within the token, though it still falls short when scaling to full syntactic structures due to its inability to capture long-range contextual dependencies and global sentence-level grammar. The pipeline and its theoretical framing are structured as follows:

1. **Mathematical Formalization:** Let $\mathbf{x} = (x_1, x_2, \dots, x_T)$ denote the sequence of characters forming an unnormalized input token, and let $\mathbf{y} = (y_1, y_2, \dots, y_T)$ be the corresponding sequence of predicted transformation labels. The linear-chain CRF models the conditional probability $P(\mathbf{y}|\mathbf{x})$ as:

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp \left(\sum_{t=1}^T \sum_k w_k f_k(y_t, y_{t-1}, \mathbf{x}, t) \right)$$

where $f_k(y_t, y_{t-1}, \mathbf{x}, t)$ represents a feature function capturing character context and label transitions, w_k is the learned weight associated with f_k , and $Z(\mathbf{x})$ is the partition function ensuring a valid probability distribution across all possible label paths:

$$Z(\mathbf{x}) = \sum_{\mathbf{y}'} \exp \left(\sum_{t=1}^T \sum_k w_k f_k(y'_t, y'_{t-1}, \mathbf{x}, t) \right)$$

2. **Alignment and Edit Operations:** The target label space $\mathcal{Y}$ is derived through optimal string alignment based on character-level Levenshtein distance, mapping source characters x_t to target characters. Each label y_t encapsulates an atomic rewrite rule, categorized into three fundamental operations:
 - **Substitution (SUB):** The source character x_t is mapped directly to a distinct target character $y_t \neq x_t$.
 - **Insertion (INS):** A target character is inserted following x_t . In the sequence labeler, this is encoded by appending the inserted character to the label of x_t (e.g., $y_t = "x_t + c"$).
 - **Deletion (DEL):** The source character x_t is removed, represented by a special null label $y_t = \epsilon$.
3. **Contextual Feature Engineering (Windowing):** The discriminative power of the CRF relies heavily on the definition of contextual feature functions f_k that model local dependencies. Instead of processing characters in isolation, a sliding context window of size w is evaluated around each position t . For each character x_t , the extracted feature vector includes:
 - **Local N-grams:** Individual characters and contiguous character sequences within the window $[t - w, t + w]$.
 - **Boundary Markers:** Explicit indicators identifying whether x_t resides at the word start (^) or word end (\$).
 - **Rule Priority Features:** Statistical indicators derived from the rule-based extraction phase, embedding historical transformation frequencies directly into the feature space.

4. **Sequence Prediction and Token Reconstruction:** During the inference phase, the system processes an unseen source string $\mathbf{x}$ by first mapping it to its respective character-level feature representation. This context is passed to the trained model using a single-instance sequence prediction method (`predict_single`) to output the globally most probable sequence of target labels $\hat{\mathbf{y}} = \arg \max_{\mathbf{y}} P(\mathbf{y}|\mathbf{x})$. Crucially, since the predicted target space may include explicit gap markers (-) inherited from the training alignment phase, a deterministic post-processing step is executed: the predicted character sequence is joined into a continuous string, and any residual gap tokens are stripped out via string replacement to synthesize the final normalized word.

4.3.4 Norm-BiLSTM: Character-Level Sequence-to-Sequence

To overcome the rigidity of hand-crafted rules and the contextual limitations of traditional sequence labelers, the primary focus of this method shifts toward deep recurrent sequence-to-sequence architectures. This definitive approach addresses both word-level and full sentence-level normalization by leveraging a bidirectional long short-term memory (BiLSTM) network with attention.

Vocabulary Representation and Dataset Structuring

To process the linguistic variations at the most granular level, the model operates on a dedicated character-level vocabulary represented by a custom class `Vocabulary`. The tokenization space is constrained to a closed set of characters directly extracted from the training observations, alongside four explicit functional tokens:

- `<PAD>`: Padding token used to equalize batch sequence lengths.
- `<SOS>`: Start-of-Sequence token prepended to target sequences to initialize autoregressive decoding.
- `<EOS>`: End-of-Sequence token appended to terminate sequence generation.
- `<UNK>`: Out-of-Vocabulary token reserved for unseen character symbols.

The raw input pairs are loaded through a custom PyTorch dataset pipeline, which maps the original and normalized strings into parallel numerical vectors of character indices. To enable efficient mini-batch processing without loss of positional fidelity, sequences are dynamically padded to the maximum length within each batch using a custom collation function (`collate_fn`), returning both the padded numerical tensors and their corresponding true, unpadded lengths.

Sequence-to-Sequence Encoder-Decoder Architecture

The core of the deep learning approach relies on a Sequence-to-Sequence architecture, which decouples the representation of the input string from the generation of the target normalized word or sentence. This formalization addresses a fundamental limitation of traditional sequence labeling models (such as CRFs), which are mathematically constrained to a strict, synchronized alignment. In contrast, the Seq2Seq method models the normalization task as an asymmetric transduction, enabling the mapping of an input sequence of length N to a target sequence of length M . This flexibility is essential for capturing natural language phenomena where characters are compressed, expanded, or omitted during the normalization process.

The architecture is divided into two main computational blocks:

- **The Encoder:** The encoder consists of a bidirectional LSTM network. Given an input sequence of character embeddings $\mathbf{E}_{in} = (e_1, e_2, \dots, e_N)$, the forward LSTM reads the sequence from left to right ($\vec{h}_i$), while the backward LSTM reads it from right to left ($\overleftarrow{h}_i$). The two hidden states are concatenated at each time step i to form a comprehensive contextual representation $h_i = [\vec{h}_i; \overleftarrow{h}_i] \in \mathbb{R}^{2d}$, where d represents the hidden state dimension. To bridge the gap between the encoder and decoder, these representations are projected through a non-linear activation to initialize the decoder state.
- **The Decoder:** Unlike the encoder, the decoder operates in a strictly unidirectional, autoregressive manner. At each step t , it predicts the next character probability distribution conditioned on its internal hidden state s_t , the previous target character embedding, and a context vector c_t supplied by the attention mechanism, continuing until the End-of-Sequence (<EOS>) token is emitted.

Character-Level Alignment via Bahdanau Attention

The most critical component of the sequence-to-sequence model is the character-level attention mechanism, which resolves the information bottleneck of traditional encoder-decoder architectures. In low-resource orthographic normalization, target characters often correspond directly to specific source character clusters, though the alignment may shift dynamically due to insertions or contractions. This architecture implements an additive (Bahdanau) [2] attention mechanism.

Let $h_i \in \mathbb{R}^{2d}$ represent the bidirectional representation of the i -th character in the source sequence of length N . Let s_t represent the hidden state of the decoder at step t .

To handle architectures where the decoder contains multiple layers ($L > 1$), the multi-layer hidden states are first flattened and projected into a single representation $s'_t \in \mathbb{R}^d$:

$$s'_t = \tanh(W_{\text{proj}} \cdot [s_{t,1}; \dots; s_{t,L}] + b_{\text{proj}}) \quad (4.1)$$

where $[s_{t,1}; \dots; s_{t,L}] \in \mathbb{R}^{L \cdot d}$ represents the concatenation of the hidden states across all layers, $W_{\text{proj}} \in \mathbb{R}^{d \times (L \cdot d)}$ is the learnable projection matrix and $b_{\text{proj}} \in \mathbb{R}^d$ is its corresponding bias vector. If $L = 1$, s'_t matches the single-layer hidden state.

At each step t , the alignment engine measures the compatibility of each encoder output h_i with the current projected decoder state s'_t . The alignment energy $e_{t,i}$ is calculated as:

$$e_{t,i} = v^\top \tanh(W_a \cdot [s'_t; h_i] + b_a) \quad (4.2)$$

where $W_a \in \mathbb{R}^{d \times 3d}$ represents the weight matrix of the alignment layer, $b_a \in \mathbb{R}^d$ is its bias vector, and $v \in \mathbb{R}^d$ is the output projection vector.

These raw energy scores are normalized across the temporal dimension of the source string using a softmax function to obtain the attention weights $\alpha_{t,i}$:

$$\alpha_{t,i} = \frac{\exp(e_{t,i})}{\sum_{j=1}^N \exp(e_{t,j})} \quad (4.3)$$

The dynamic context vector c_t is then computed as the weighted sum of the encoder outputs:

$$c_t = \sum_{i=1}^N \alpha_{t,i} h_i \quad (4.4)$$

This context vector c_t is concatenated with the target character embedding at step t and passed as input to the decoder LSTM, allowing the network to explicitly focus on relevant source character sub-sequences during transduction.

4.3.5 Data Augmentation Strategies

One of the most critical challenges when processing low-resource languages is data scarcity. To alleviate this limitation and prevent overfitting, expanding the training dataset via data augmentation is highly beneficial. For the character-level normalization task, two distinct data augmentation methodologies were developed.

Rule-Based Denormalization

The first data augmentation strategy operates by inverting the deterministic rules extracted during the rule-based phase (Section 4.3.2). While the rule-based framework can be deployed directly for text normalization, its inverted rules are also leveraged to perturb clean sentences to simulate human-generated typographical noise, abbreviations, and dialectal variations. Given a dataset of standardized text without corresponding non-standardized equivalents, this inverse process generates synthetic input-target pairs, expanding the available training data for machine learning approaches such as the CRF model described in Section 4.3.3.

Back Normalization

In contrast, the second approach is designed to expand the dataset for the neural sequence-to-sequence architecture (Section 4.3.4) through an auxiliary, specular neural model. Instead of introducing a different architecture, this method leverages the exact same bidirectional sequence-to-sequence network with attention described in Section 4.3.4. However, the model is trained in a reversed configuration: it learns to map clean, standardized sentences back to their raw, non-standardized forms. By training this inverse transduction system, we process larger monolingual standardized corpora to synthesize corresponding noisy inputs for the primary normalization model.

4.4 Experimental Setup

To ensure a rigorous and reproducible evaluation of the normalization task, the hyperparameter configurations for each of the three proposed approaches were systematically calibrated. All experiments were executed using the computing infrastructure detailed in Appendix A. Given the structural and mathematical differences inherent to each approach, ranging from deterministic rule extraction to probabilistic sequence modeling and deep neural architectures, each framework required a distinct optimization strategy to balance computational efficiency with sequence-level accuracy.

The following sections detail the specific parameter spaces, tuning methodologies, and final optimal configurations selected for each method.

4.4.1 Norm-RB: Deterministic Rule-Based Extraction

For the rule-based framework (Section 4.3.2) and its corresponding inverted augmentation engine (Section 4.3.5), the optimization configuration governs the offline synthetic generation process rather than direct inference. To prevent combinatorial explosion, maintain phonetic alignment, and balance downstream dataset expansion for the CRF model, the following optimal hyperparameters were enforced during rule application and generation:

- **Context Window Sizes (w):** Rules were extracted across three distinct character-level context boundaries ($w \in \{1, 2, 3\}$) to maximize structural and contextual diversity.
- **Maximum String Operations:** Restricted to a maximum of 2 character-level perturbations (substitutions, deletions, or insertions) per individual token.
- **Maximum Morphological Variations:** Capped at a maximum of 2 distinct synthetic variants generated from a single original clean lemma.
- **Minimum Confidence Threshold:** Filtered the inverted knowledge base to retain only transformation rules exhibiting an extraction confidence score of $c \geq 0.1$.

4.4.2 Norm-CRF: Conditional Random Fields

The training and optimization of the CRF sequence labeler (Section 4.3.3) rely on the L-BFGS optimization algorithm (`algorithm='lbfgs'`), capped at a maximum of 250 iterations to guarantee numerical convergence. To ensure robustness against unseen label sequences during inference, the model incorporates all possible state transitions (`all_possible_transitions=True`), computing penalty weights even for unobserved training pairs.

To prevent overfitting and control model complexity, a randomized hyperparameter search was executed across 20 iterations with a 3-fold cross-validation scheme (60 total training runs). The optimization grid targeted the regularization parameters c_1 (L_1 Lasso regularizer) and c_2 (L_2 Ridge regularizer). Both parameters were sampled mathematically from an exponential distribution profile (`scipy.stats.expon`), expressed as:

$$c_1, c_2 \sim \text{Expon}(\text{scale} = 0.1)$$

where the scale parameter is configured to 0.1 to focus the randomized search densely around smaller, highly-effective regularization constraints. The objective function maximized the weighted F_1 -score to account for class imbalance within the character-level transitions.

Following the exploration phase, the optimal configuration yielded a regularizer value of $c_1 = 0.3088$ and $c_2 = 0.1068$. These specific parameters define the final optimized estimator used for downstream evaluation tasks, balancing feature sparsity with architectural stability.

4.4.3 Norm-BiLSTM: Character-Level Sequence-to-Sequence

The deep learning architectures developed for both word-level and sentence-level sequence-to-sequence normalization (Section 4.3.4) were systematically optimized using a comprehensive grid search strategy.

Hyperparameter Search Space

The search space targeted the architectural and optimization constraints of the sequence-to-sequence model across the following dimensions:

- **Embedding and Hidden Dimensions:** The character embedding layer maps input tokens into a continuous vector space, while the hidden state size defines the capacity of the recurrent cells to capture long-term historical dependencies.
- **Network Depth and Regularization:** The number of recurrent layers determines the hierarchical abstraction capability of the network. A dropout regularization rate is evaluated to prevent overfitting by randomly zeroing out hidden activations between subsequent recurrent layers.
- **Optimization Schedule:** The learning rate governs the step size during gradient descent via the Adam optimizer, operating across a fixed batch size and a maximum number of training epochs.
- **Convergence and Generation:** An early stopping mechanism is enforced to monitor validation loss using a specific patience threshold (consecutive epochs without improvement) to prevent overtraining. Concurrently, the teacher forcing ratio dictates the probability of feeding the ground-truth token rather than the model’s own prediction back into the decoder at each time step.

Optimal Architecture and Execution Constraints

Following the grid search phase, the optimal configuration selected for the final evaluation tasks is summarized in Table 4.4. During the exploration, scaling the network depth from 1 to 2 layers yielded no statistically significant performance improvements; thus, a single-layer configuration was prioritized to reduce computational overhead. Consequently, although a dropout rate of 0.3 was originally evaluated, it was structurally deactivated in the final model since dropout applies exclusively between subsequent recurrent layers (i.e., when depth > 1).

Furthermore, the mini-batch size was strictly constrained to 8 instances. This conservative allocation was explicitly selected to mitigate hardware memory pressures stemming from long contextual text sequences and to ensure stable execution within a shared, multi-user high-performance computing environment.

Hyperparameter	Optimal Value
Character Embedding Dimension	128
Hidden State Size	512
Number of Recurrent Layers	1
Dropout Rate	0.0 (<i>Deactivated due to Depth = 1</i>)
Learning Rate	1×10^{-3}
Batch Size	8
Maximum Training Epochs	100
Early Stopping Patience	15
Teacher Forcing Ratio	1.0

Table 4.4: Optimal hyperparameter configuration selected for the Norm-BiLSTM via grid search optimization.

4.5 Experimental Results and Discussion

The models are evaluated on the test set partitions described in Section 4.2.2 using the Character Error Rate (CER) as the primary performance metric. To guarantee algorithmic consistency across all methods, the metric is computed via the standardized **CharErrorRate** module provided by the `torchmetrics.text` framework.

Formally, the CER measures the edit distance between the model’s predicted sequence and the clean target sequence at the character level. It calculates the minimum number of character-level operations, specifically substitutions (S), deletions (D), and insertions (I), required to transform the predicted string into the reference gold standard, normalized by the total number of characters in the reference sequence (N):

$$CER = \frac{S + D + I}{N}$$

A lower CER percentage indicates higher normalization accuracy, with 0% representing a flawless sequence replication.

The CER represents the most suitable metric for the normalization task because target transformations typically affect only localized character sequences rather than the entire token structure. Consequently, this character-level granularity provides a less punitive and more accurate reflection of model performance compared to rigid word-level evaluation frameworks.

Crucially, within the scope of this evaluation, the performance profiles of models operating exclusively on isolated word tokens are considered secondary objective functions. Evaluating models strictly at the word level disregards critical contextual and syntactic dynamics, failing to capture adjacent linguistic markers such as articles, clitics, or specific verbal conjugations. Consequently, the ultimate empirical milestone of this framework is not merely to minimize the CER on the word-level test set, but rather to optimize accuracy across the full contextual boundaries of the sentence-level test partition.

Nevertheless, maintaining a low error rate on isolated words remains an important operational asset, as it ensures the final normalizer can gracefully process standalone user queries or decoupled out-of-vocabulary lemmas. This dual operational requirement justifies the hybrid configuration of the definitive training dataset, which intentionally aggregates both structural word-level lemmas and full contextual sentence-level instances.

4.5.1 Norm-RB: Deterministic Rule-Based Extraction

The rule-based framework described in Section 4.3.2 initially parses the clean-to-unclean alignment pairs across different context window sizes (w) to extract deterministic transformation patterns.

Qualitative Rule Analysis

The parsing routine successfully populated the knowledge base, yielding a highly structured set of morphological rules. A qualitative selection of the most frequent and confident transformation rules extracted from the training dataset using a context window of $w = 1$ is summarized in Table 4.5.

op	src	dst	left	right	freq.	conf.	Example ($S \rightarrow T$)
SUB	ó	o	i	n	332	0.9970	abitaçión → abitaçion
SUB	a	â	i	\$	178	0.9834	addescia → addesciâ
SUB	a	â	t	\$	169	1.0000	abbasta → abbastâ
DEL	-		n	a	152	1.0000	a-a sestrinn-a → aa sestrinna
DEL	n		ñ	a	132	1.0000	a-a sestrinn-a → a-a sestriña
INS		g	g	i	117	0.9915	agiàdda → aggiadda

Table 4.5: Quantitative distribution of the top six deterministic transformation rules extracted from the dataset using a context window size of $w = 1$.

When evaluated as a standalone inference model, preliminary testing revealed severe operational limitations: due to the strict rigidity of exact-match context windows, the deterministic model failed to generalize to unseen lexical forms, resulting in fragmented outputs and high error rates. Consequently, the method was pivoted: the framework was repurposed exclusively as an offline denormalization engine (Section 4.3.5) to generate synthetic training instances for probabilistic estimators.

Synthetic Expansion for Norm-CRF

The data augmentation strategy was deployed exclusively to expand the training partition for the Norm-CRF pipeline. By inverting the extracted transformation rules across all three context windows ($w \in \{1, 2, 3\}$), 2106 alignment pairs were successfully generated under strict quality constraints (maximum of 2 string operations per token, maximum 2 synthetic variants per lemma, minimum confidence $c \geq 0.1$). This process expanded the baseline CRF training dataset from 14597 to an aggregate size of 16703 instances.

Extraction Window Size (w)	Generated Synthetic Pairs
Window size $w = 1$	1557
Window size $w = 2$	384
Window size $w = 3$	165
Total Augmented Subset	2106

Table 4.6: Quantitative distribution of the synthetic alignment pairs generated for the CRF training set, categorized by extraction window size.

4.5.2 Norm-CRF: Conditional Random Fields

The empirical evaluation of the Norm-CRF framework yielded stable performance boundaries on the word-level test partition. When trained exclusively on the baseline dataset, the optimized CRF model achieved a CER of 5.07%. Incorporating the rule-based synthetically augmented dataset resulted in a minor performance shift, yielding a CER of 5.14%.

Training Configuration	Training Set Size	Word-Level CER (%) ↓
Word-Level Dataset (Section 4.2.1)	14597	5.07%
Augmented Dataset (Section 4.5.1)	16703	5.14%

Table 4.7: Comparative performance of the Norm-CRF framework on the word-level test partition across training configurations.

For a traditional machine learning architecture operating strictly under localized character-level features, these scores represent a robust baseline. The marginal variance (+0.07% CER) indicates that the deterministic data perturbation mechanism introduced minimal noise into the probabilistic transition matrices while significantly increasing training feature density.

4.5.3 Norm-BiLSTM: Character-Level Sequence-to-Sequence

To evaluate the sequence-to-sequence Bidirectional LSTM architecture (Section 4.3.4), the empirical investigation examines how different granularities of training data, ranging from isolated words to full contextual sequences, influence character-level normalization accuracy across varied evaluation environments.

The model was benchmarked across both test set partitions (1817 isolated words and 381 contextual sentences described in Section 4.2.2) under four distinct training methods:

- **Norm-BiLSTM_word**: Trained exclusively on isolated word tokens.
- **Norm-BiLSTM_sentence**: Trained solely on sentence-level contextual sequences.
- **Norm-BiLSTM_2step**: Sequential curriculum learning (initial training on words, followed by fine-tuning on sentences).
- **Norm-BiLSTM_mix**: Trained simultaneously on the joint aggregation of word-level and sentence-level data.

Configuration	Word-Level Test CER (%) ↓	Sentence-Level Test CER (%) ↓
Norm-BiLSTM_word	3.09%	57.00%
Norm-BiLSTM_sentence	14.59%	4.41%
Norm-BiLSTM_2step	4.86%	5.40%
Norm-BiLSTM_mix	2.59%	4.44%

Table 4.8: Baseline performance comparison across Norm-BiLSTM training methods on word-level (1817 samples) and sentence-level (381 samples) test partitions.

As detailed in Table 4.8 and illustrated in Figure 4.1, Norm-BiLSTM_sentence achieves the lowest error on sentences (4.41%), but degrades severely on isolated words (14.59%) due to out-of-vocabulary bottlenecks. Conversely, the joint configuration Norm-BiLSTM_mix emerges as the most versatile architecture: it achieves the top performance on isolated words (2.59%) while remaining within a negligible 0.03% margin of Norm-BiLSTM_sentence on full sentences (4.44%). The higher error of Norm-BiLSTM_2step on words (4.86%) confirms that sequential fine-tuning suffers from catastrophic forgetting. Thus, Norm-BiLSTM_mix is selected as the primary baseline.

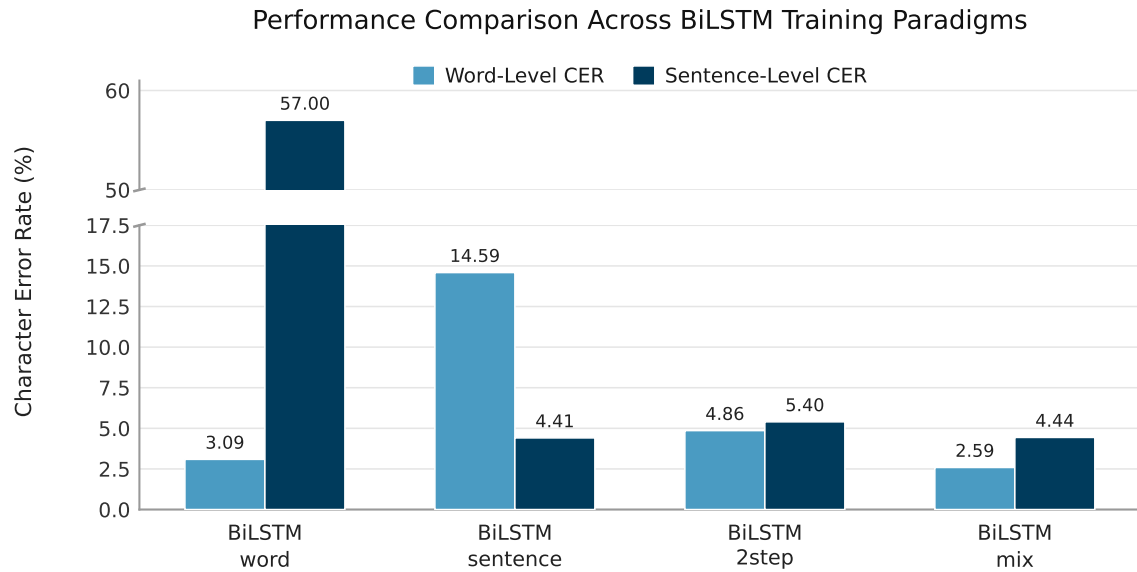


Figure 4.1: Comparative performance of Norm-BiLSTM training methods on Word-Level vs. Sentence-Level test partitions.

Sequence Length Constraints and Bottlenecks

When deploying Norm-BiLSTM architectures for character-level text normalization, an essential aspect of the empirical analysis concerns whether input sequence length introduces a critical performance bottleneck and at what specific threshold recurrent error accumulation causes accuracy to degrade. As sequence lengths increase, recurrent hidden states naturally suffer from information bottlenecks, attention fragmentation, and compounding error propagation during autoregressive decoding.

An initial diagnostic analysis confirmed that sentence length distributions were uniformly balanced across the Training, Validation, and Testing partitions, as illustrated in Figure 4.2. This structural balance ensured consistent sequence complexity during both optimization and evaluation while preventing hardware memory saturation.

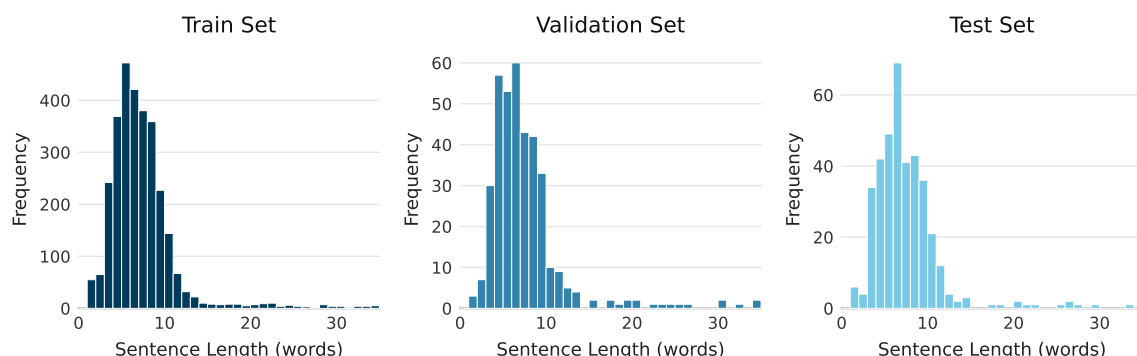


Figure 4.2: Sentence length distribution across Training, Validation, and Testing partitions, highlighting structural balance across all dataset splits.

However, a granular breakdown of the validation error profiles unveiled a severe architectural bottleneck when correlating CER with sentence length. Grouping validation sentences into six distinct word-length clusters (1-5, 6-10, 11-15, 16-20, 21-30, and 31-50 words) revealed a sharp performance degradation threshold beyond 20 words, as depicted in Figure 4.3.

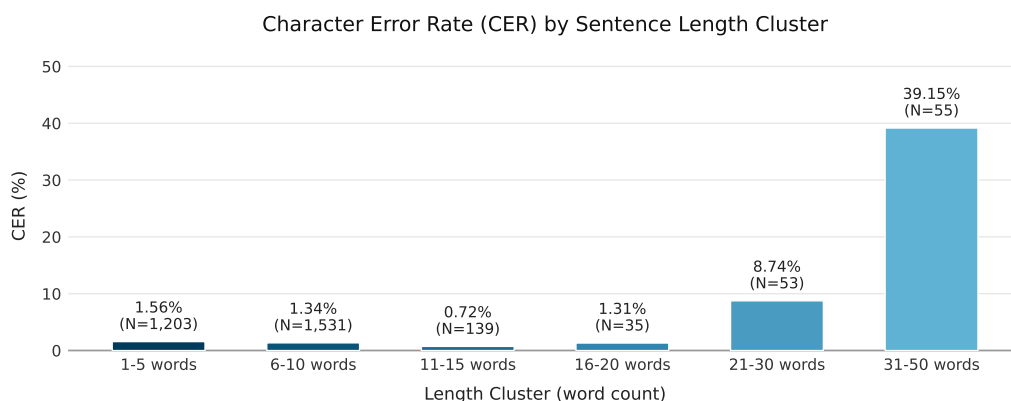


Figure 4.3: Progression of Character Error Rate (CER) across sentence length clusters, highlighting critical performance degradation beyond the 20-word threshold.

Although the architecture achieves optimal performance on medium sequences of 11-15 words with a minimum CER of 0.72%, the segmentation limit was deliberately set to 20 words (1.31% CER) to allow the model to accommodate moderately long sentences while maintaining an acceptable error rate. Beyond this threshold, performance degrades rapidly: average CER escalates to 8.74% in the 21-30 word interval, ultimately collapsing to an unsatisfactory 39.15% on long sequences containing between 31 and 50 words.

Structural Sentence Splitting Strategies

To address the severe length-induced degradation identified above, the study investigates whether dynamic sequence segmentation can effectively mitigate long-range memory bottlenecks without destroying global sentence context or introducing artificial syntactic errors. A data-driven sequence segmentation strategy was engineered around the empirical inflection point at the 20-word boundary, implementing two distinct splitting approaches:

- **Hard Splitting:** A deterministic truncation strategy where any sequence exceeding 20 words is strictly segmented exactly at the 20th word boundary, regardless of syntactic structure.
- **Soft Splitting:** A semantically guided segmentation strategy that scans sentences longer than 20 words for structural punctuation landmarks (such as commas, semicolons, or periods) near the target threshold to preserve local clause integrity. If no valid punctuation mark is present, the pipeline gracefully falls back to the deterministic hard split.

Beyond stabilizing the recurrent hidden states, both splitting mechanisms expanded the total sequence count across all dataset partitions while preserving the structural integrity of the original data splits. The quantitative breakdown of these newly restructured corpora is summarized in Table 4.9.

Dataset Variant	Total	Train	Validation	Test
Sentence-Level Dataset	3791	3030	380	381
Hard-Split Dataset	3987	3190	401	396
Soft-Split Dataset	4047	3242	409	396

Table 4.9: Quantitative dimensions of sentence-level dataset partitions across original, hard-split, and soft-split configurations.

Re-training and evaluating the `Norm-BiLSTM_mix` baseline on these length-constrained datasets successfully mitigated the recurrent degradation bottleneck. As detailed in Table 4.10, the empirical outcomes indicate that length-constrained corpora dramatically improve sentence normalization, reducing the CER from 4.44% down to 1.55% on soft-split contexts.

Furthermore, while the soft-split test partition allows all model variants to achieve superior normalization accuracy overall, the model trained on the hard-split dataset (**Norm-BiLSTM_mix_hard**) demonstrates a marginal advantage when evaluated across both split test suites (1.91% hard, 1.52% soft). To thoroughly investigate this trade-off, both splitting methodologies were retained for further benchmarking in conjunction with the synthetic back-normalization pipeline.

Dataset Configuration	Word-Level CER (%) ↓	Sentence-Level Test CER (%) ↓		
		Original	Hard-Split	Soft-Split
Norm-BiLSTM_mix	2.59%	4.44%	1.98%	1.55%
Norm-BiLSTM_mix_hard	2.77%	4.24%	1.91%	1.52%
Norm-BiLSTM_mix_soft	2.64%	4.88%	2.09%	1.57%

Table 4.10: Performance of Norm-BiLSTM_mix evaluated on original vs. length-constrained dataset configurations.

Impact of Back-Normalization

To address data scarcity in low-resource contextual training, the final evaluation phase explores the extent to which generating synthetic non-standard text via a reverse normalization pipeline can overcome dataset limitations and systematically reduce sequence-level error rates.

Applying the reverse-normalization pipeline (Section 4.3.5) across native *grafia unităia* contextual text sequences yielded an expanded dataset of 55055 training entries (37428 back-normalized sentences aggregated with 17627 baseline instances). The proportional composition of this definitive unified dataset is illustrated in Figure 4.4.

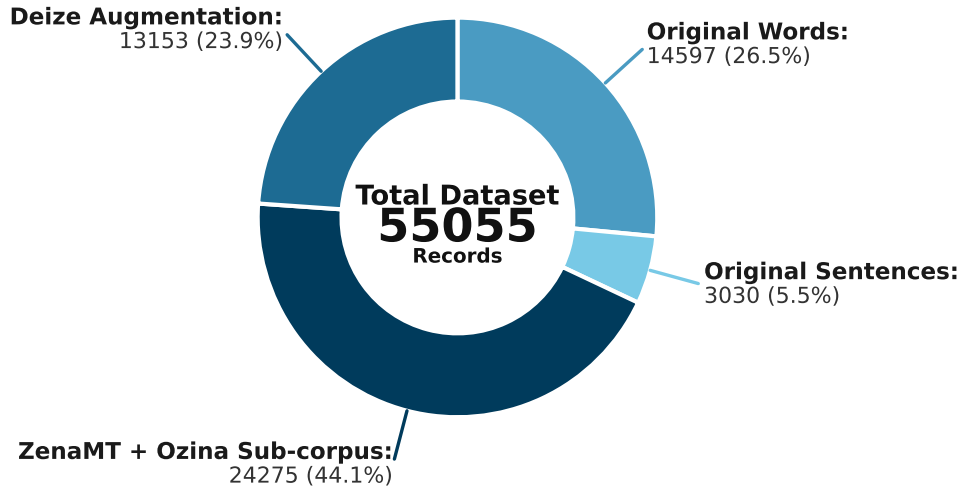


Figure 4.4: Proportional composition of the definitive unified Norm-BiLSTM training dataset.

As detailed in Table 4.11, integrating back-normalized synthetic data systematically outperforms all previous configurations. The Norm-BiLSTM_mix_syn_hard architecture reaches optimal sentence normalization accuracy across all evaluation suites (1.33% CER on soft-split sequences).

Model Architecture	Word-Level CER (%)	Sentence-Level Test CER (%) ↓		
		Original	Hard-Split	Soft-Split
Norm-BiLSTM_mix	2.59%	4.44%	1.98%	1.55%
Norm-BiLSTM_mix_syn_soft	2.83%	4.76%	1.85%	1.34%
Norm-BiLSTM_mix_syn_hard	2.62%	4.40%	1.79%	1.33%

Table 4.11: Performance comparison between baseline and synthetic-augmented Norm-BiLSTM models.

4.5.4 Comparative Analysis and State-of-the-Art Benchmark

To contextualize overall gains, the proposed architectures are benchmarked against the state-of-the-art baseline proposed by Lusito, Maillard and Ferrante [36] in Table 4.12.

Model Architecture	Test CER (%) ↓	
	Word-Level	Sentence-Level
Maillard et al. [36]	—	2.47%
Norm-CRF	5.07%	—
Norm-BiLSTM_mix	2.59%	1.55%
Norm-BiLSTM_mix_syn_soft	2.83%	1.34%
Norm-BiLSTM_mix_syn_hard	2.62%	1.33%

Table 4.12: Comprehensive performance comparison across proposed variants and the state-of-the-art baseline on Genoese text normalization.

The empirical findings demonstrate that the proposed **Norm-BiLSTM_mix_syn_hard** framework substantially outperforms the state-of-the-art benchmark, reducing the sentence-level CER from 2.47% down to 1.33%, as visually summarized in Figure 4.5. Combining length-constrained sequence segmentation with back-normalized synthetic augmentation successfully mitigates both small-sample constraints and long-range recurrent memory degradation.

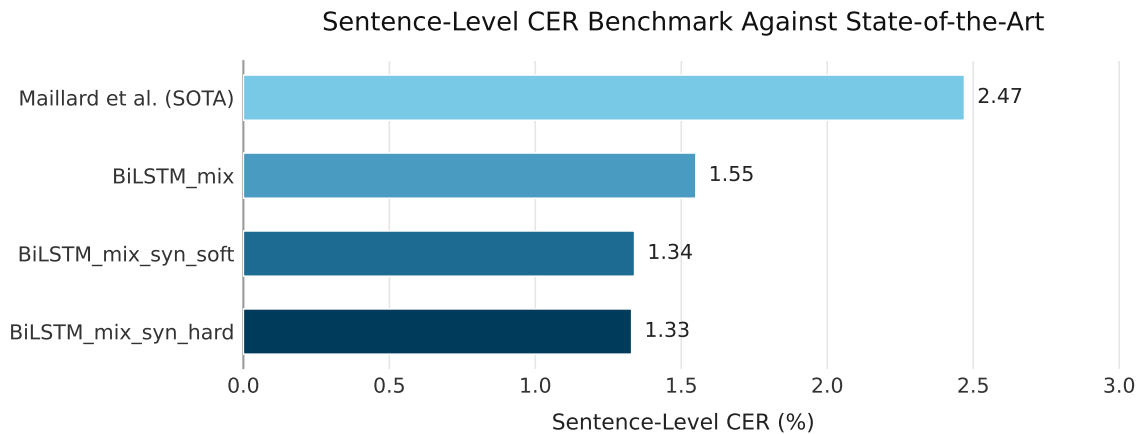


Figure 4.5: Sentence-level CER progression across experimental phases compared to the State-of-the-Art baseline by Maillard et al.

4.5.5 Qualitative Analysis

This section presents a qualitative breakdown of normalized sentences extracted from the short split of the test set, evaluated using the optimal model configuration identified in the previous benchmark, `Norm-BiLSTM_mix_syn_hard`. To systematically analyze model behavior, predictions are categorized into three distinct performance tiers based on their Character Error Rate (CER).

Tier 1: High Precision Normalization		
Ex. 1	Source:	Ėse ä dispoziōn d'ûnn-a persōnn-a
	Target:	Ėse a-a dispoziōn de unna persoña
	Pred:	Ėse a-a dispoziōn de unna persoña
Ex. 2	Source:	Sensa ûn pō de fortûnn-a nō se fa ninte,
	Target:	Sensa un pō de fortuna no se fa ninte,
	Pred:	Sensa un pō de fortuna no se fa ninte,
Ex. 3	Source:	Ciêuve a derrûo
	Target:	Ceuve à derruo
	Pred:	Ceuve à derruo

Table 4.13: Qualitative examples demonstrating flawless character-level transduction.

Table 4.13 illustrates instances of flawless normalization (Tier 1). In these cases, the model accurately executes multi-character structural transformations, successfully resolving complex diacritics and non-standard orthographic patterns, mapping ‘*persōnn-a*’ to the tilde-based ‘*persoña*’, while correctly expanding hyphenated prepositional contractions ‘*ä*’ → ‘*a-a*’. Furthermore, Example 3 highlights the model’s ability to handle complex vocalic shifts and accentuation rules, correctly normalizing the archaic diphthong ‘*iêu*’ into ‘*eu*’ in ‘*Ciêuve*’ → ‘*Ceuve*’, inserting the required grave accent on the preposition ‘*à*’, and removing the circumflex long-vowel marker in ‘*derrûo*’ → ‘*derruo*’.

Tier 2: Moderate Discrepancies

Ex. 4	Source: Quando $\bar{o}$ pöveo $\bar{o}$ fa limöxina a-o ricco $\bar{o}$ diao $\bar{o}$ se ne rie, Target: Quande o pöveo o fa limöxina a-o ricco o diao o se ne rie, Pred: Quando o pöveo o fa limöxina a-o ricco o diao o se ne rie,
Ex. 5	Source: e finanze de quest'opea pia se trêuvan in condizioîn ciüttosto critiche. Target: e finanze de quest'opea pia se treuvan in condiçioin ciutòsto critiche. Pred: e finanze de quest'opea pia se treuvan in condiçioin ciuttòsto critiche.
Ex. 6	Source: Nisciûn a láschia anâ pe sò camín, Target: Nisciun â lascia anâ pe sò cammin, Pred: Nisciun a lascia anâ pe sò cammin,

Table 4.14: Qualitative examples illustrating minor local errors.

In Table 4.14, Tier 2 highlights moderate discrepancies where the output is largely correct but suffers from localized errors. In Example 4, while the model successfully eliminates all non-standard macron vowels ‘ $\bar{o}$ ’ $\rightarrow$ ‘ o ’, it fails to apply the final-vowel reduction in ‘*Quando*’, retaining the source form instead of producing the expected target ‘*Quande*’. In Example 5, the model over-generalizes consonant gemination by generating a double *t* in ‘*ciüttòsto*’ rather than the standardized single-consonant form ‘*ciutòsto*’. Finally, Example 6 demonstrates an accentuation and clitic contraction oversight: although the model correctly standardizes the geminate ending ‘*camín*’ $\rightarrow$ ‘*cammin*’ and removes the acute accent ‘*láschia*’ $\rightarrow$ ‘*lascia*’, it fails to produce the clitic object pronoun in its contracted form with a circumflex accent ‘*â*’ incorrectly predicting ‘*a*’ instead.

Tier 3: High Error Cases	
Ex. 7	Source: Sciòrto mi cö mæ röbon,
	Target: Sciòrto mi co-o mæ röbon
	Pred: Sciòrto mi co-o ræbon
Ex. 8	Source: Da bagnâse ûn pö l'ezzo ff ago!
	Target: Da bagnâse un pö l'esòfago!
	Pred: Da bagnâse un pö l' ezzo ffago!
Ex. 9	Source: Ciù de Cholera o de Grippa,
	Target: Ciù de Còlera ò de Grippa,
	Pred: Ciù de Cholera o de Grippa,

Table 4.15: Qualitative examples highlighting high-error predictions, characterized by character-level hallucinations.

Finally, Table 4.15 reports Tier 3 samples exhibiting higher CER values. These degradations primarily stem from out-of-vocabulary or low-frequency lexical items that the architecture struggles to map accurately. In Example 7, a character-level fusion error occurs, where features from the preceding possessive pronoun ‘*mæ*’ are erroneously hallucinated into the noun root while dropping the pronoun entirely, yielding ‘*ræbon*’ instead of ‘*mæ röbon*’. In Example 8, when encountering rare or archaic phonetic spellings in ‘*ezzo**ff**ago*’, the model defaults to a passive strategy, leaving the raw input unchanged instead of performing the required graphemic transformation into ‘*esòfago*’. Similarly, Example 9 illustrates a passive copy failure on foreign or archaic graphies and disjunctive conjunctions: the model retains the latinized spelling ‘*Cholera*’ without introducing the grave accent ‘*Còlera*’ and misses the open-vowel accentuation on the disjunctive conjunction ‘*o*’ → ‘*ò*’.

Overall, the qualitative evaluation confirms that the model achieves robust performance across standard syntax and high-frequency patterns, while performance degradation is predominantly localized to highly unfamiliar or archaic lexical structures.

Chapter 5

Translation

Machine translation is the computational task of automatically converting textual content from a source language into a target language while preserving both semantic meaning and stylistic nuance. Driven by recent advancements in neural architectures, machine translation for high-resource languages, such as English, Spanish, or Italian, has achieved near human fluency and high translation quality.

However, applying these methodologies to low-resource and regional languages like Genoese presents severe computational challenges. Due to the scarcity of parallel training data (as discussed in Chapter 3), the presence of historical orthographic fragmentation, and limited domain-specific corpora, standard neural translation models frequently struggle with severe vocabulary sparsity, hallucinations, and poor syntactic generalization.

Deploying neural machine translation in such constrained settings requires addressing several critical hurdles to achieve competitive results. A primary challenge involves mitigating orthographic noise and graphemic instability through the text normalization techniques detailed in Chapter 4. Beyond text standardization, overcoming data scarcity necessitates specialized modeling strategies, including subword tokenization adaptation, parameter-efficient fine-tuning, and multilingual transfer learning, which together form the core methodology of this chapter.

5.1 Related Works in Machine Translation

Machine translation for low-resource languages has experienced a paradigm shift, moving from rule-based and phrase-based statistical frameworks to deep neural architectures. However, adapting neural machine translation to regional languages like Genoese presents severe challenges due to high orthographic fragmentation and parallel data scarcity.

This section reviews the state-of-the-art architectures, pre-trained multilingual models, and parameter-efficient adaptation techniques that form the foundation of our experimental framework.

5.1.1 Multilingual Sequence-to-Sequence Architectures

In low-resource scenarios, training deep NMT models entirely from scratch often leads to severe overfitting. To mitigate data scarcity, recent literature heavily relies on cross-lingual transfer learning via pre-trained sequence-to-sequence architectures. These models leverage vast multilingual corpora during pre-training to build shared representations that can be fine-tuned for specialized machine translation tasks.

Prominent architectures employed in low-resource research include:

- **MarianNMT:** Proposed by Junczys-Dowmunt et al. [27], MarianNMT is an efficient C++-based framework and family of encoder-decoder Transformer models pre-trained on diverse bilingual and multilingual pairs, providing lightweight baselines for domain adaptation and fine-tuning.
- **mBART-50:** Proposed by Tang et al. [60], mBART-50 is a multilingual sequence-to-sequence denoising autoencoder pre-trained on large-scale monolingual corpora across 50 languages using the BART objective. Although Genoese is absent from mBART-50’s native pre-training pool, the model provides a rich morphological and syntactic representation space due to its extensive coverage of closely related Romance languages (e.g., Italian, French, Spanish, and Portuguese). Research in cross-lingual transfer learning demonstrates that unrepresented target languages can be adapted by expanding the pre-trained vocabulary and initializing new language indicator tokens from typologically similar source languages, allowing the model to reuse its pre-trained representations for supervised fine-tuning.
- **NLLB (No Language Left Behind):** Designed by Meta [14], NLLB-200 represents the current State-of-the-Art (SOTA) in massively multilingual machine translation, scaling direct machine translation across more than 200 languages without relying on English as a pivot. NLLB-200 natively includes Genoese (accessible via the language tag `lij_Latn`), having been pre-trained on web-mined regional text, primarily from Genoese Wikipedia.

Architecturally, deep multilingual models like NLLB adopt a standard Transformer Encoder-Decoder framework:

- **Encoder Stack:** Processes the source-language sequence into continuous hidden representations. Each encoder layer consists of multi-head self-attention mechanisms and position-wise feed-forward networks, mediated by pre-layer normalization and residual connections.
- **Decoder Stack:** Generates the target-language sequence autoregressively. Each decoder layer incorporates a cross-attention sub-layer positioned between causal self-attention and the feed-forward network, conditioning target generation on the source representations.

While pre-trained massively multilingual models provide strong off-the-shelf baselines, native pre-training on raw web-mined text leaves them vulnerable to non-standard orthographic variations, necessitating domain-specific fine-tuning on normalized parallel corpora.

5.1.2 Generative LLMs and Token-Free Models

In contrast to traditional encoder-decoder architectures, modern state-of-the-art machine translation increasingly leverages **decoder-only generative LLMs** operating as causal language models. While sequence-to-sequence models explicitly decouple source encoding from target decoding via cross-attention, causal generative models process translation within a single, unified Transformer stack.

Subword tokenization algorithms (e.g., byte-pair encoding [55]) can sometimes encounter limitations in low-resource settings due to out-of-vocabulary expansion and non-standard orthographies. To address this issue, Xue et al. [70] proposed **ByT5**, a token-free model operating directly on raw byte sequences. By processing character and byte-level representations, ByT5 exhibits superior robustness to spelling variations, noisy text, and dialectal variations without requiring specialized subword vocabularies.

Concurrently, recent open-weight initiatives focus on expanding regional and low-resource European language coverage. Proposed by Martins et al. [38], the **EuroLLM** project provides a suite of open-weight multilingual language models specifically engineered to address linguistic diversity across European languages. Pre-trained on a dense multilingual corpus covering all 24 official languages of the European Union alongside regional minority languages (such as Catalan), EuroLLM provides a rich latent space of shared morphosyntactic features applicable to the Gallo-Italic continuum.

A key structural advantage of EuroLLM stems from its subword tokenization space. Utilizing byte-pair encoding trained jointly across its multilingual corpus, EuroLLM features an expanded vocabulary of 128000 tokens. This significantly reduces the **fertility rate** (the average number of subwords generated per word) for Romance languages, preventing sequence fragmentation and maximizing context-window efficiency during autoregressive generation.

5.1.3 Parameter-Efficient Fine-Tuning Strategies

Full parameter fine-tuning of multi-billion parameter pre-trained models (such as NLLB-200 or EuroLLM) presents prohibitive computational overhead and high GPU memory (VRAM) consumption during backpropagation. To address these resource constraints without sacrificing cross-lingual representation quality, PEFT methodologies have become the standard paradigm to prevent catastrophic forgetting.

Low-Rank Adaptation (**LoRA**), proposed by Hu et al. [23], freezes the pre-trained model parameter matrices $W_0 \in \mathbb{R}^{d \times k}$ and injects trainable rank decomposition matrices into the Transformer layers. The parameter update ΔW is constrained by factorizing it into two low-rank matrices, $B \in \mathbb{R}^{d \times r}$ and $A \in \mathbb{R}^{r \times k}$, where the rank $r \ll \min(d, k)$:

$$h = W_0x + \Delta Wx = W_0x + \frac{\alpha}{r}BAx \quad (5.1)$$

where x represents the input vector and α is a constant scaling hyperparameter that stabilizes optimization across different values of r . Matrix A is typically initialized from a Gaussian distribution, while matrix B is initialized to zero, ensuring $\Delta W = 0$ at the onset of training.

Quantized Low-Rank Adaptation (**QLoRA**), proposed by Dettmers et al. [15], further enhances memory efficiency by quantizing the base model’s frozen weights down to 4-bit precision while optimizing high-precision Low-Rank Adapters, making the adaptation of large architectures feasible on consumer hardware. QLoRA relies on three core technical pillars:

1. **4-bit NormalFloat:** An information-theoretically optimal quantile quantization data type designed for normally distributed weights, yielding lower quantization error than standard 4-bit data types.
2. **Double Quantization:** A secondary quantization step applied to the quantization constants themselves, compressing the memory footprint further without sacrificing model precision.

3. **Paged Optimizers:** A memory management technique leveraging CUDA Unified Memory to handle gradient spikes during backpropagation by dynamically paging optimizer states across CPU and GPU memory.

Beyond weight-based adapters, advanced intervention methodologies such as Representation Fine-Tuning (**ReFT**) and LoReFT, proposed by Wu et al. [69], shift the adaptation focus from weight updates to editing hidden representations directly during training. By intervening on a small fraction of hidden state representations, ReFT achieves competitive performance with significantly fewer trainable parameters than standard parameter-efficient methods.

5.1.4 Recent Advances in Low-Resource Language Adaptation

Beyond classical transfer learning and back-translation, recent literature has explored novel paradigms to optimize generative capabilities, vocabulary representations, and multimodal integration for low-resource languages.

To overcome severe data scarcity, advanced data augmentation and synthetic generation strategies have gained significant traction. Early foundational work by Fadaee et al. [18] introduced a targeted data augmentation method for low-resource NMT by replacing frequent words in source sentences with rare words in plausible contexts using language models, thereby significantly improving the model’s ability to translate low-frequency vocabulary. In morphologically complex low-resource settings, synthetic generation offers targeted training signals, for instance, Yildirim-Erbasli and Bulut [72] demonstrated that domain-tailored data augmentation techniques in Turkish substantially improve downstream text generation and reading comprehension tasks. However, generating high-quality synthetic text in low-resource regimes remains challenging due to model hallucinations. To address this, Cegin et al. [10] introduced activation steering, an inference-time intervention mechanism that directs the internal representations of large language models to synthesize higher-fidelity, structurally coherent data for data-scarce target languages.

In parallel, addressing subword fragmentation during tokenization remains a primary bottleneck for machine translation. Rather than relying on static, high-resource tokenizers that artificially inflate sequence lengths, Naik et al. [42] proposed targeted vocabulary expansion strategies. By dynamically extending the pre-trained embedding space with domain- and language-specific tokens, this approach mitigates out-of-vocabulary degradation and improves translation quality across low-resource translation pairs.

Finally, the deployment of large language models and large multimodal models in regional language contexts has been extensively evaluated. Assessing the capabilities of foundational models on non-Western languages, Mgeni [39] systematically benchmarked LLMs on Swahili, highlighting key insights into prompt sensitivity, fine-tuning efficiency, and cross-lingual generalization limits in low-resource African languages. Extending these architectures beyond textual data, Lupaşcu et al. [34] provided a comprehensive survey on large multimodal models for low-resource languages, demonstrating how grounding textual representations with speech and visual modalities can stabilize model transfer and compensate for missing parallel text.

5.1.5 Machine Translation for the Genoese Language

Early research into computational resources for Genoese focused primarily on dictionary compilation, rule-based morphosyntactic processing, and initial parallel text alignment. Pioneer studies by Rosso and Buscaldi laid the groundwork for processing Ligurian regional variants, highlighting the impact of non-standardized orthography on computational tools. Their work emphasized how dialectal fragmentation creates severe out-of-vocabulary bottlenecks, hindering traditional statistical machine translation approaches.

With the evolution of deep learning, recent literature has shifted focus toward evaluating and specialized training of NMT architectures for Genoese. Although massively multilingual models natively support the language, most notably Meta’s NLLB-200 [14] via the `lij_Latn` tag, their off-the-shelf performance remains problematic.

In a dedicated evaluation, Buscaldi and Rosso [7] constructed a novel Italian-Genoese test dataset to systematically benchmark NLLB-200 across translation directions. Their study revealed severe limitations, particularly when translating into Genoese, demonstrating that training pipelines reliant predominantly on raw web-mined text and Wikipedia suffer from significant hallucination, structural errors, and orthographic inconsistencies.

To address the limitations of web-mined data, Haberland et al. [20] introduced **ZenaMT**, a novel, culturally relevant Italian-Genoese parallel dataset manually curated to reflect authentic language use. By training a compact Transformer-based model directly on this high-quality corpus, they achieved significant performance gains over Meta’s massively multilingual NLLB-200 model, proving that dataset quality and domain alignment can outweigh raw parameter scale in ultra-low-resource scenarios.

5.2 Data Processing for Automatic Translation

Unlike traditional machine translation pipelines operating on high-resource language pairs, the translation phase for Genoese requires specialized preprocessing to handle structural noise and orthographic inconsistencies inherent to low-resource language datasets.

To address orthographic fragmentation across heterogeneous sources, preprocessing prior to translation relies on an **automatic text normalization** step. All Genoese texts are systematically routed through the automated normalization module detailed in Chapter 4. This step homogenizes non-standard, historical, or regional spelling variants into a unified target orthography (*grafia unitäia*), effectively reducing vocabulary sparsity and out-of-vocabulary rates before the parallel data is fed into the translation models.

5.2.1 Translation Dataset

Following normalization, dataset construction involves selecting, structuring, and filtering available resources across different granularities:

- **Sentence-Level Parallel Dataset:** Bilingual sentence pairs are filtered and aligned to construct the primary parallel dataset for direct supervised context-aware translation experiments.
- **Word-Level Lexical Pairs:** Standalone bilingual lemmas and dictionary entries are extracted to provide broad lexical coverage and enrich vocabulary mapping.
- **Monolingual Target Dataset:** Authentic monolingual Genoese texts are curated to serve as input for the back-translation data augmentation pipeline (Section 2.2.3).

5.2.2 Dataset Splitting and Dimensions

To evaluate the generalization capacity and robustness of the translation framework, the parallel datasets were partitioned into distinct subsets tailored to cross-lingual learning.

The machine translation framework relies on the sentence-level parallel dataset introduced in Section 5.2. In addition, to investigate the impact of broader lexical coverage, specific experimental configurations leverage a *Mixed* paradigm, where the entire bilingual word-level dataset is appended exclusively to the training partition. This strategy enriches vocabulary representation without corrupting the validation and test distributions with isolated non-contextual tokens.

A deterministic random splitting strategy was enforced using a fixed seed to guarantee exact experimental reproducibility. Alignment pairs in the sentence-based translation dataset were partitioned according to a standard 80%/10%/10% ratio for the Training, Validation, and Test sets, respectively. The quantitative breakdown of these partitions across all experimental configurations is summarized in Table 5.1.

Dataset Variant	Total Records	Train	Validation	Test
Word-Level	23482	23482	–	–
Sentence-Level	30831	24664	3083	3084
Mixed	54313	48146	3083	3084

Table 5.1: Quantitative distribution of records across experimental data partitions for the machine translation task.

5.3 Methods

Building upon the socio-linguistic context and data scarcity challenges discussed in Chapter 2, automating translation for Genoese serves both as a complex technical benchmark and as a concrete vehicle for digital language preservation. Due to the high orthographic fragmentation of raw sources, the translation framework relies strictly on the automated normalization module (Chapter 4) as a mandatory prerequisite step, ensuring that all input variants are homogenized into *grafia unitäia* prior to translation.

To systematically evaluate the machine translation task, this section outlines four distinct methodological approaches, spanning from compact architectures trained from scratch to high-capacity pre-trained models:

- **Custom BiLSTM:** A lightweight sequence-to-sequence model trained entirely from scratch on subword representations.
- **mBART-50:** Cross-lingual transfer learning using a pre-trained multilingual sequence-to-sequence model adapted via target vocabulary extension.
- **NLLB-200:** Supervised fine-tuning and zero-shot evaluation of Meta’s massively multilingual NMT architecture.
- **EuroLLM:** Parameter-efficient fine-tuning of a generative, decoder-only Large Language Model optimized for European languages.

5.3.1 Task Formulation and Subword Segmentation

Unlike the character-level transduction framework employed for orthographic normalization, machine translation between Standard Italian and Genoese is formalized as a sequence-to-sequence mapping at the **subword token level**.

Formally, given an input source sequence $X = (x_1, x_2, \dots, x_N)$ in Standard Italian and a corresponding target sequence $Y = (y_1, y_2, \dots, y_M)$ in normalized Genoese (*grafia unitäia*), the primary objective is to learn a conditional probability distribution $P(Y|X)$. The translation system autoregressively predicts each target subword token y_t conditioned on the source sequence X and the previously generated target tokens $y_{<t}$:

$$P(Y|X) = \prod_{t=1}^M P(y_t \mid y_{<t}, X) \quad (5.2)$$

The transition from character-level to subword-level representations is driven by fundamental structural differences between normalization and translation:

- **Morphological Transduction vs. Semantic Shift:** While orthographic normalization operates on minor graphemic variations where the underlying lexical root remains identical, translation maps distinct lexical items and syntactic structures across two languages.
- **Sequence Length Constraints:** Operating at the character level for full translation drastically expands sequence lengths (N, M) , introducing severe computational overhead and compounding autoregressive decoding errors.
- **Vocabulary Sparsity:** Strict word-level tokenization would require an excessively large vocabulary matrix in a low-resource setting like Genoese, leading to severe out-of-vocabulary rates.

To balance character-level flexibility with word-level semantic richness, subword segmentation is implemented using **Byte-Pair Encoding (BPE)**. The algorithm iteratively decomposes rare or inflected words into recurring subword units while preserving frequent words as unified tokens. For Genoese, this strategy allows models to explicitly share structural features across languages, isolating common prefixes, suffixes, and verbal roots to optimize parameter efficiency.

5.3.2 NMT-BiLSTM: Subword-Level Sequence-to-Sequence

To establish direct architectural continuity with the orthographic normalization pipeline, the first approach utilizes a sequence-to-sequence model trained entirely from scratch. Specifically, the network relies on a Bidirectional LSTM encoder coupled with a unidirectional LSTM decoder, mediated by an additive Bahdanau attention mechanism.

While the underlying recurrent engine and attention formulation remain structurally identical to the normalization network, the core distinction lies in tokenization and dataset management:

- **Byte-Level BPE Tokenization:** Rather than processing character sequences, the subword vocabulary is built using a BPE algorithm augmented with a byte-level pre-tokenizer (**ByteLevel**). This processes spaces and punctuation natively at the byte level without requiring pre-splitting heuristics, ensuring seamless decoding through a matching **ByteLevelDecoder**.
- **Shared Vocabulary and Functional Tokens:** Functional tokens (**<PAD>**, **<SOS>**, **<EOS>**, **<UNK>**) manage dynamic mini-batching and autoregressive generation. Target sequences are prepended with **<SOS>** and appended with **<EOS>**, while source sequences append **<EOS>**. The vocabulary space is unified across both languages, resulting in a shared embedding matrix $E \in \mathbb{R}^{|\mathcal{V}| \times d_e}$ scaled to a target BPE size $|\mathcal{V}| = 10000$.

Deploying this custom recurrent architecture serves a dual purpose: testing whether a compact, non-pretrained model can scale to cross-lingual subword translation when supported by normalized text, and establishing an essential baseline against which the performance gains of transfer learning and generative LLMs can be benchmarked.

5.3.3 NMT-mBART: Unseen Language Adaptation

The second translation approach leverages **mBART-50**, a large-scale multilingual sequence-to-sequence denoising autoencoder [60], to evaluate cross-lingual transfer learning. Although Genoese is absent from mBART-50’s native pre-training pool, the model provides a rich morphological and syntactic foundation due to its extensive coverage of typologically related Romance languages (such as Italian, French, and Portuguese). The foundational design of this fine-tuning strategy was explored during preliminary course research and has been structurally refined within this thesis to form a core component of the unified comparative framework.

Because Genoese is not supported natively, the adaptation methodology incorporates three main architectural and training strategies:

Vocabulary Expansion and Control Token Initialization

To enable sequence-to-sequence conditioning, a new language-specific control token, `<lij>`, is added to the tokenizer, and the base embedding layer matrices are expanded from $|\mathcal{V}_{\text{mBART}}|$ to $|\mathcal{V}_{\text{mBART}}| + 1$. To evaluate the impact of latent semantic proximity on model convergence, three warm-start initialization strategies are benchmarked for the `<lij>` token vector:

- **Italian Initialization (`<it_IT>`):** Copies weights directly from the Italian token embedding, establishing the primary baseline due to maximal typological and lexical proximity.
- **French Initialization (`<fr_XX>`):** Copies weights from the French token embedding to test whether shared Gallo-Italic phonetic and lexical affinities facilitate decoder alignment.
- **Portuguese Initialization (`<pt_XX>`):** Copies weights from the Portuguese token embedding to assess cross-lingual transfer from a related Western Romance language.

Two-Phase Curriculum Fine-Tuning Strategy

Inspired by the principles of Curriculum Learning [57], the training process gradually increases task complexity to optimize transfer learning and prevent severe initial gradient disruption:

1. **Phase 1 (Lexical Foundation):** The model is trained exclusively on isolated dictionary pairs. This phase acts as a lexical anchor, forcing the model to align Italian and Genoese embeddings at the lexical level and reducing initial generation errors on core terminology.
2. **Phase 2 (Contextual Fluency and Memory Replay):** The model transitions to learning full sentence structures. To prevent catastrophic forgetting of the terminology acquired in Phase 1, a memory-replay strategy is introduced by mixing a fixed percentage of dictionary pairs directly into the narrative sentence batches.

Quantized Low-Rank Adaptation

To enable full sequence-to-sequence backpropagation while mitigating computational and memory constraints, adaptation relies on Quantized Low-Rank Adaptation [15]. The underlying mBART-50 base parameters are frozen and quantized to 4-bit NormalFloat precision, while trainable low-rank adapter matrices are injected into the Transformer attention layers, constraining parameter updates to a high-precision low-rank subspace.

5.3.4 NMT-NLLB: Native Language Adaptation

The third approach leverages Meta’s **No Language Left Behind (NLLB-200)**, a massively multilingual Sequence-to-Sequence transformer architecture capable of direct translation across more than 200 languages. Unlike mBART-50, NLLB-200 natively includes Genoese (designated via the control token `lij_Latn`), having been pre-trained on web-mined regional text.

Within our translation pipeline, NLLB-200 serves a dual operational role:

- **Zero-Shot Baseline:** Evaluated directly without parameter updates to measure off-the-shelf translation quality on unnormalized vs. normalized Genoese text.
- **Supervised Fine-Tuning Backbone:** Subjected to domain-specific fine-tuning on our normalized *grafia unitäia* parallel dataset to align its internal generations with the target orthographic standard.

To overcome VRAM limitations during full-parameter backpropagation on multi-billion parameter Transformer stacks, optimization is conducted using Low-Rank Adaptation (LoRA). Trainable rank decomposition matrices (A and B) with rank $r = 16$ and scaling factor $\alpha = 32$ are injected into the cross-attention and self-attention projections, keeping the original pre-trained weights frozen.

5.3.5 NMT-EuroLLM: Decoder-Only Generative Model

The final paradigm evaluates **EuroLLM**, a decoder-only generative Large Language Model operating as a Causal Language Model. Unlike encoder-decoder architectures, EuroLLM processes translation within a single Transformer stack, formulating generation as an instruction-following sequence completion task.

EuroLLM was selected due to its pre-training design, which explicitly targets European language coverage (including Standard Italian, French, Catalan, and Portuguese). Its native 128,000-token BPE vocabulary significantly reduces subword fertility rates for Romance languages, preserving context-window efficiency.

To adapt EuroLLM for Italian-to-Genoese translation, the fine-tuning methodology combines two core technical components:

- **Instruction Prompting and Completion-Only Loss Masking:** Source sentences and target translations are formatted into structured prompts. During back-propagation, Cross-Entropy loss is calculated strictly on the generated target subwords in *grafia unitäia*, while instruction prompts and source sentence tokens are masked using a dummy label index (-100).
- **Quantized Low-Rank Adaptation :** Parameter adaptation is conducted by quantizing the base model’s frozen weights to 4-bit NormalFloat precision while training high-precision Low-Rank Adapters injected into the linear query, key, value, and output projection layers.

5.3.6 Data Augmentation via Back-Translation

To mitigate severe parallel data scarcity, a standard Back-Translation data augmentation pipeline is integrated across the experimental framework. While acquiring human-aligned parallel sentences is challenging, authentic target-side monolingual Genoese texts are more readily available.

Following standard neural machine translation augmentation practices, the Back-Translation workflow operates in three steps:

1. **Reverse Model Training:** A secondary target-to-source model (Genoese $\rightarrow$ Italian) is trained on the baseline parallel dataset.
2. **Synthetic Generation:** The reverse model translates an unaligned dataset of authentic monolingual Genoese sentences into synthetic Italian text.
3. **Dataset Integration:** The resulting synthetic pairs (synthetic Italian source, authentic Genoese target) are merged with the primary parallel dataset to train the forward translation models (Italian $\rightarrow$ Genoese).

By employing Back-Translation exclusively, the forward models are exposed to authentic, naturally written Genoese target text, preserving native dialectal nuances and vocabulary while scaling the parallel training volume.

5.4 Experimental Setup

To ensure a rigorous and reproducible evaluation of the machine translation task, the hyperparameter configurations for each of the four neural models were systematically calibrated. All experiments were executed using the high-performance computing infrastructure detailed in Appendix A. Given the distinct structural properties inherent to each computational paradigm, ranging from recurrent sequence-to-sequence networks to massively multilingual transformer encoder-decoders and causal decoder-only LLMs, each model class required a tailored training strategy to optimize cross-lingual transfer, convergence speed, and parameter efficiency.

The following sections detail the specific parameter spaces, optimization methodologies, and final hyperparameter configurations selected for each translation paradigm.

5.4.1 NMT-BiLSTM: Subword-Level Sequence-to-Sequence

To ensure structural consistency with the orthographic normalization pipeline (Section 4.4.3), the sequence-to-sequence translation model preserves the core dimensional and optimization engine of the recurrent baseline while adapting specific hyperparameters to accommodate subword-level cross-lingual translation.

Hyperparameter Selection and Architectural Continuity

The parameter space was systematically calibrated through empirical exploration to balance cross-lingual representation capacity against the computational constraints of the parallel dataset. During preliminary tuning, multiple configurations for embedding dimensions ($d_e \in \{64, 128, 256\}$) and hidden state capacities ($d_h \in \{128, 256, 512\}$) were evaluated, among these, maintaining $d_e = 128$ and $d_h = 512$ provided the optimal trade-off between sequence-level translation accuracy and convergence stability. Preserving these core dimensions allows direct experimental continuity with the normalization model while ensuring identical gradient dynamics and VRAM consumption patterns:

- **Core Layer Dimensions:** The embedding layer maps tokens into a continuous vector space of dimension 128, while the hidden state capacity of the recurrent cells is maintained at 512 units to capture long-term context.
- **Optimization Engine:** Optimization relies on the Adam algorithm with an initial learning rate of 1×10^{-3} and a mini-batch size of 8 instances, ensuring consistent optimization behavior across tasks.

- **Generation and Alignment:** Decoded target sequences are supervised using cross-entropy loss with an explicit ignore index for padding tokens and a teacher forcing ratio of 1.0 during training execution.

Optimal Model and Task-Specific Adjustments

Despite these shared foundational properties, shifting from character-level orthographic normalization to subword-level machine translation required three key architectural adjustments derived from systematic hyperparameter testing. First, the vocabulary size for byte-pair encoding was evaluated across a range from 10000 to 20000 tokens, a shared vocabulary size of $|\mathcal{V}| = 15000$ subword units yielded the best translation quality by minimizing out-of-vocabulary instances while avoiding excessive subword fragmentation. Second, exploring network depth ($N \in \{1, 2, 3\}$) demonstrated that scaling from 1 to 2 recurrent layers ($N = 2$) was essential to model complex inter-lingual morphosyntactic alignments. With depth > 1 , a dropout rate of 0.3 was activated between subsequent LSTM layers, supplemented by an L_2 weight decay of 1×10^{-4} to prevent overfitting. Finally, the early stopping patience threshold was reduced to 5 consecutive epochs to avoid over-adaptation to synthetic or small-scale parallel sequences.

The complete hyperparameter configuration selected for the translation NMT-BiLSTM framework is summarized in Table 5.2.

Hyperparameter	Optimal Value
Vocabulary Type & Size ($ \mathcal{V} $)	BPE (15000 tokens)
Embedding Dimension (d_e)	128
Hidden State Size (d_h)	512
Number of Recurrent Layers (N)	2
Dropout Rate	0.3
Weight Decay (L_2)	1×10^{-4}
Learning Rate	1×10^{-3}
Batch Size	8
Maximum Training Epochs	100
Early Stopping Patience	5
Teacher Forcing Ratio	1.0

Table 5.2: Optimal hyperparameter configuration for the NMT-BiLSTM.

5.4.2 NMT-mBART: Unseen Language Adaptation

It should be noted that the fine-tuning phase for the mBART-50 model was originally conducted as part of a preliminary course project on a separate workstation using unnormalized source data. To maintain methodological consistency within this thesis, the final model checkpoint evaluation was re-executed using the standardized evaluation dataset described in Section 5.2.1 and the high-performance computing environment detailed in Appendix A. Throughout all training runs, optimization parameters were strictly calibrated to ensure stability across both translation directions ($\text{ITA} \rightarrow \text{LIJ}$ and $\text{LIJ} \rightarrow \text{ITA}$).

Execution Schedule and Data Allocation

Fine-tuning execution strictly adheres to the two-phase schedule defined in Section 5.3.3:

- **Phase 1 Execution:** Executed for 5 full training epochs using 100% of the processed *TIG* dictionary dataset. Optimization employs a primary learning rate of 1×10^{-4} .
- **Phase 2 Execution:** Executed for a maximum of 20 macro-epochs on the complete parallel sentence dataset. Batches are supplemented with a 15% random sample of the *TIG* dictionary pairs for lexical replay. For the $\text{ITA} \rightarrow \text{LIJ}$ translation direction, training data is further expanded using synthetic parallel sentences generated via back-translation.

Optimization and Convergence Constraints

Gradient updates are computed using the AdamW optimizer with a linear learning rate scheduler and a weight decay of 0.01. Base model parameters are loaded in 4-bit NormalFloat precision via QLoRA. Validation loss is monitored after every epoch, enforcing an early stopping patience threshold of 5 consecutive epochs without loss improvement to avoid overtraining.

The complete hyperparameter configuration for the NMT-mBART setup is summarized in Table 5.3.

Hyperparameter	Optimal Configuration / Value
Base Model Checkpoint	facebook/mbart-large-50-many-to-many-mmt
Target Language Token	<lij> (<i>Added & Expanded</i>)
Fine-Tuning Paradigm	QLoRA (4-bit NormalFloat Quantization)
Optimizer	AdamW
Learning Rate	1×10^{-4}
Weight Decay	0.01
Phase 1 Duration & Dataset	5 Epochs (100% <i>TIG</i> Lexical Pairs)
Phase 2 Duration & Dataset	20 Epochs (Sentence Dataset + 15% <i>TIG</i> Replay)
Early Stopping Patience	5 Epochs

Table 5.3: Hyperparameter setup and execution constraints for fine-tuning NMT-mBART on Genoese machine translation.

5.4.3 NMT-NLLB: Native Language Adaptation

To evaluate state-of-the-art massively multilingual transfer learning, the third machine translation pipeline relies on Meta’s **NLLB-200** model [14]. Experiments evaluate both off-the-shelf **zero-shot translation** and **parameter-efficient fine-tuning** using the distilled 600M parameter checkpoint (**facebook/nllb-200-distilled-600M**). Unlike mBART-50, NLLB-200 natively supports Genoese via the **lij_Latn** language tag alongside Italian (**ita_Latn**).

Parameter-Efficient Adaptation

To fine-tune the model within GPU VRAM constraints while preventing catastrophic forgetting, Parameter-Efficient Fine-Tuning is implemented via Low-Rank Adaptation in half-precision floating-point format (**float16**). Low-rank adapter matrices are injected directly into the attention projections of the transformer model. The adapter rank is set to $r = 16$ with a scaling factor $\alpha = 32$, a LoRA dropout rate of 0.05, and trainable bias parameters deactivated.

Optimization Constraints and Execution Schedule

Input and target sequences undergo BPE tokenization capped at a maximum length of 128 subword tokens. Model optimization is driven by the AdamW algorithm with an initial learning rate of 5×10^{-4} , a weight decay of 0.01, and a batch size of 16 instances per GPU device. Fine-tuning runs for 6 macro-epochs, with model evaluation performed at the end of each epoch to checkpoint and save exclusively the model instance achieving the lowest validation loss (`eval_loss`).

The full hyperparameter setup for the NMT-NLLB adaptation framework is summarized in Table 5.4.

Hyperparameter	Optimal Configuration / Value
Base Model Checkpoint	facebook/nllb-200-distilled-600M
Source / Target Language Tags	ita_Latn (Italian), lij_Latn (Genoese)
Execution Paradigms	Zero-Shot Baseline & QLoRA Fine-Tuning
Precision	16-bit Floating Point (fp16)
Maximum Sequence Length	128 Subword Tokens
LoRA Rank (r) & Alpha (α)	$r = 16$, $\alpha = 32$
LoRA Dropout Rate	0.05
Learning Rate	5×10^{-4}
Weight Decay	0.01
Batch Size	16
Maximum Training Epochs	6
Model Selection Metric	Best Validation Loss (<code>eval_loss</code>)

Table 5.4: Hyperparameter configuration and execution constraints for fine-tuning NMT-NLLB on Genoese machine translation.

5.4.4 NMT-EuroLLM: Decoder-Only Generative Model

To evaluate generative causal language modeling, the fourth machine translation pipeline relies on **EuroLLM-9B-Instruct** (`utter-project/EuroLLM-9B-Instruct`) [38]. Operating as a 9-billion parameter decoder-only model, EuroLLM is adapted to translation tasks using **supervised fine-tuning** formatted with ChatML instruction templates.

System Prompt Formatting and Template Alignment

Dataset instances are structured into conversational turns using EuroLLM’s native ChatML template schema. Each sample encapsulates a direction-specific system instruction, the source context as the **user** prompt, and the target reference as the **assistant** response:

- **Italian to Genoese (ITA $\rightarrow$ LIJ):** *‘Traduci la seguente frase dall’italiano al ligure genovese rispettando rigorosamente la grafia unitäia.’*
- **Genoese to Italian (LIJ $\rightarrow$ ITA):** *‘Traduci la seguente frase dal ligure genovese (in grafia unitäia) all’italiano corretto.’*

Grid Search Optimization and QLoRA Configuration

To determine the optimal LoRA hyperparameter combination across translation directions, a systematic grid search was executed over 6 distinct experimental configurations. The search space evaluated rank sizes $r \in \{16, 32, 64\}$ with scaling factors $\alpha = 2r$, learning rates $\eta \in \{1 \times 10^{-4}, 3 \times 10^{-4}\}$, and a constant LoRA dropout rate of 0.05. Optimization was monitored using early stopping with a patience of 2 evaluation epochs based on validation loss (`eval_loss`).

The grid search identified directional parameter divergences:

- **ITA $\rightarrow$ LIJ Optimal Setup:** $r = 64$, $\alpha = 128$, and $\eta = 3 \times 10^{-4}$, reflecting the need for higher rank capacity when generating low-resource target text.
- **LIJ $\rightarrow$ ITA Optimal Setup:** $r = 32$, $\alpha = 64$, and $\eta = 3 \times 10^{-4}$, providing sufficient capacity for decoding into the high-resource target language.

Base parameters were frozen and quantized to 4-bit NormalFloat precision (**nf4**) with double quantization, utilizing **bfloat16** compute precision and SDPA attention optimization. Trainable LoRA adapters were injected across all linear projections of the transformer model.

SFT Training Dynamics

Fine-tuning was conducted using the **SFTTrainer** module from the TRL library. Training executed over 4 macro-epochs with a per-device batch size of 4 and 4 gradient accumulation steps (yielding an effective global batch size of 16). Input sequences were padded on the left to a maximum length of 256 tokens with 100 linear warmup steps.

The optimal hyperparameter configuration for the NMT-EuroLLM fine-tuning setup is summarized in Table 5.5.

Hyperparameter	Optimal Configuration / Value
Base Model Checkpoint	<code>utter-project/EuroLLM-9B-Instruct</code>
Instruction Format	ChatML Template (<code>system</code> , <code>user</code> , <code>assistant</code>)
Quantization Data Type	4-bit NormalFloat (<code>nf4</code>) with Double Quantization
Compute Precision	<code>bfloat16</code> (<code>sdpa</code> Attention)
LoRA Rank (r) & Alpha (α) (ITA $\rightarrow$ LIJ)	$r = 64, \alpha = 128$
LoRA Rank (r) & Alpha (α) (LIJ $\rightarrow$ ITA)	$r = 32, \alpha = 64$
LoRA Dropout Rate	0.05
Learning Rate	3×10^{-4}
Per-Device Batch Size / Grad. Accum.	4 / 4 (Global Batch Size = 16)
Warmup Steps	100
Maximum Sequence Length	256 Tokens (Left-Padding)
Maximum Fine-Tuning Epochs	4 Epochs
Grid Search Selection Strategy	Best Validation Loss (<code>eval_loss</code>) with Early Stopping

Table 5.5: Optimal hyperparameter setup and grid search selection for fine-tuning NMT-EuroLLM on Genoese machine translation.

5.5 Experimental Results and Discussion

To comprehensively assess the performance of the four proposed translation models, all final models are evaluated across three distinct datasets. The primary benchmark consists of the held-out test split extracted from our curated parallel dataset, detailed in Section 5.2.1. To evaluate cross-domain generalization under standardized conditions, models are additionally benchmarked on two widely adopted external datasets: **FLORES+** [6] and **Bouquet** [61], both integrated directly via Hugging Face. Crucially, all intermediate experimental variations, hyperparameter tuning steps, and validation checks were conducted exclusively using the custom validation set, the external benchmarks were reserved solely for the final evaluation of fully trained checkpoints. Consequently, intermediate comparative analyses in this section report metrics on the custom test split.

Quantitative performance is measured across a set of complementary evaluation metrics, where standard metrics are computed across all evaluations, while specialized measures such as MetricX, GlotLID score, and the adjusted **Glottlid_x_metric** are exclusively utilized within the official Hugging Face leaderboard for the Bouquet benchmark:

- **BLEU (Bilingual Evaluation Understudy)** [47]: Evaluates n -gram precision between candidate translations and reference texts, incorporating a brevity penalty to prevent overly short outputs:

$$\text{BLEU} = \text{BP} \cdot \exp \left(\sum_{n=1}^N w_n \log p_n \right)$$

where p_n denotes modified n -gram precision, w_n represents uniform weights ($\frac{1}{N}$ with $N = 4$), and BP is the brevity penalty defined as:

$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}$$

with c representing the candidate output length and r the effective reference length.

- **chrF++ (Character n -gram F-score with Word n -grams)** [50, 51]: Computes character-level precision and recall augmented with word-level n -grams, providing a morphological-aware metric that is particularly effective for highly inflected low-resource languages:

$$\text{chrF++} = \frac{(1 + \beta^2) \cdot P \cdot R}{\beta^2 \cdot P + R}$$

where P and R represent the combined character and word n -gram precision and recall, respectively, and $\beta = 2$ assigns higher weight to recall.

- **TER (Translation Edit Rate) [56]:** Quantifies the minimum number of edit operations (insertions, deletions, substitutions, and word shifts) required to transform a hypothesis sentence into a reference sentence, normalized by the reference length:

$$\text{TER} = \frac{\text{Number of Edits}}{\text{Average Number of Reference Words}}$$

Unlike BLEU and chrF++, a lower TER score indicates higher translation quality.

- **MetricX [29]:** An error-based metric developed by Google that evaluates translation quality by predicting a penalty score based on semantic and structural errors, where lower values indicate higher quality:

$$\text{MetricX} = f_{\theta}(\text{source}, \text{hypothesis}, \text{reference})$$

where f_{θ} is a fine-tuned Transformer-based regression model (such as an mT5 architecture) trained to mirror human multidimensional quality metrics annotations, clipped to a standard range of $[0, 25]$.

- **GlottLID score [30]:** Quantifies the language identification confidence for the generated hypothesis text, ensuring the output conforms to the expected target language script and distribution:

$$\text{GlottLID} = P(l_{\text{target}} \mid \text{hypothesis})$$

where P represents the classification probability assigned by the language identification model to the target language l_{target} .

- **GlottLID-adjusted metric (**Glottlid_x_metric**):** A composite evaluation score that scales or weights the primary quality metric (such as MetricX or chrF++) by the language identification confidence, penalizing generations that drift into unintended languages:

$$\text{Glottlid_x_metric} = \text{Metric} \cdot P(l_{\text{target}} \mid \text{hypothesis})$$

ensuring that outputs with poor target-language alignment receive appropriately adjusted evaluation scores.

Relying strictly on automated numerical metrics provides an incomplete picture in low-resource machine translation scenarios. Because surface-level metrics enforce exact n -gram overlaps, valid lexical synonyms, regional orthographic variations, or syntactically correct alternative phrasings are often heavily penalized despite being semantically accurate. Therefore, similar to the methodology adopted in the normalization chapter, this quantitative evaluation is supplemented with a detailed qualitative analysis of sample model outputs to assess contextual fluency, lexical fidelity, and morphological adequacy.

5.5.1 NMT-BiLSTM: Subword-Level Sequence-to-Sequence

Inherited directly from the orthographic normalization framework, the NMT-BiLSTM model serves as a controlled baseline to evaluate how effectively a compact, non-pretrained recurrent neural network trained strictly from scratch can scale to cross-lingual machine translation.

Deploying a lightweight recurrent model introduces a fundamental trade-off in low-resource translation settings. Unlike large transformer models, compact models do not require massive amounts of data to converge, making them remarkably sample-efficient under data sparsity. However, this structural simplicity severely restricts their overall representational capacity, rendering them highly susceptible to over-adapting to in-domain patterns and hindering generalization to out-of-domain text. Furthermore, while character-level orthographic normalization maintains sequence length symmetry between source and target tokens, cross-lingual translation introduces significant lexical and syntactic variations (as detailed in Section 2.1.4). Because translation alters word boundaries and phrase lengths unpredictably, automatic heuristics to truncate sequence lengths without breaking cross-lingual alignment cannot be applied. Resolving these discrepancies would require labor-intensive manual realignments, which yield diminishing returns for a baseline model.

In-Domain Data Composition Analysis

To evaluate the impact of training data strategies, the NMT-BiLSTM baseline was evaluated on the custom test set under standard parallel sentence training and synthetic data augmentation via backtranslation (BT) for the Italian-to-Genoese direction. Table 5.6 details the in-domain quantitative results on the custom test split.

Direction: Genoese $\rightarrow$ Italian (LIJ $\rightarrow$ ITA)			
Model Variant	BLEU $\uparrow$	chrF++ $\uparrow$	TER $\downarrow$
NMT-BiLSTM	38.46	59.16	44.98

Direction: Italian $\rightarrow$ Genoese (ITA $\rightarrow$ LIJ)			
Model Variant	BLEU $\uparrow$	chrF++ $\uparrow$	TER $\downarrow$
NMT-BiLSTM	36.46	55.28	48.21
NMT-BiLSTM-BT (Backtranslated Data)	17.42	35.25	65.36

Table 5.6: In-domain performance comparison across NMT-BiLSTM training variants on the custom test set.

On the in-domain custom test split, the standard NMT-BiLSTM achieves solid baseline scores of 38.46 BLEU for LIJ $\rightarrow$ ITA and 36.46 BLEU for ITA $\rightarrow$ LIJ. However, incorporating synthetic parallel sentences generated via backtranslation leads to a catastrophic performance degradation in the ITA $\rightarrow$ LIJ direction, causing in-domain BLEU to collapse from 36.46 to 17.42 and chrF++ to drop from 55.28 to 35.25.

This severe decline can be attributed to two compounding factors. First, it reflects the architectural capacity saturation of the compact NMT-BiLSTM model, which quickly reaches its learning limit when exposed to larger, noisier datasets. Second, because the synthetic target text was generated using the baseline Genoese-to-Italian NMT-BiLSTM model itself, the resulting parallel pairs contained accumulated translation errors and synthetic noise. Without pre-trained language modeling capabilities to filter or contextualize these inaccuracies, training on self-generated synthetic data triggers an error propagation cascade, severely impairing model convergence and generation quality.

Full Multi-Benchmark Evaluation

To evaluate out-of-domain generalization across distinct textual domains, the optimal NMT-BiLSTM configuration for each translation direction, namely the standard models trained strictly on high-purity parallel data, were benchmarked against the Bouquet and FLORES+ corpora. Table 5.7 summarizes the comprehensive multi-benchmark performance.

NMT-BiLSTM Direction: Genoese $\rightarrow$ Italian (LIJ $\rightarrow$ ITA)			
Benchmark	BLEU $\uparrow$	chrF++ $\uparrow$	TER $\downarrow$
Custom Test Set	38.46	59.16	44.98
Bouquet	13.73	38.49	79.72
FLORES+	6.19	31.53	87.63
NMT-BiLSTM Direction: Italian $\rightarrow$ Genoese (ITA $\rightarrow$ LIJ)			
Benchmark	BLEU $\uparrow$	chrF++ $\uparrow$	TER $\downarrow$
Custom Test Set	36.46	55.28	48.21
Bouquet	12.36	33.51	85.51
FLORES+	6.42	28.11	91.14

Table 5.7: Quantitative performance of the optimal standard NMT-BiLSTM configurations across all evaluation benchmarks.

The cross-benchmark empirical evaluation highlights a severe generalization gap between in-domain and out-of-domain test sets for non-pretrained models. While the **NMT-BiLSTM** baseline performs respectably on the custom split, its performance drops sharply on external datasets, collapsing to 13.73 and 12.36 BLEU on Bouquet, and reaching 6.19 and 6.42 BLEU on FLORES+ for LIJ $\rightarrow$ ITA and ITA $\rightarrow$ LIJ, respectively. This drop confirms that without pre-trained contextual representations, non-pretrained recurrent layers heavily overfit the stylistic and vocabulary distributions of their immediate training data.

Additionally, a persistent directional asymmetry is observed across all benchmarks, with translation into Italian (LIJ $\rightarrow$ ITA) consistently outperforming the reverse direction (ITA $\rightarrow$ LIJ). Decoding into Italian benefits from regular syntax and lower morphological entropy, whereas generating Genoese from scratch requires predicting complex morphological inflections without the structural guidance of a pre-trained language model prior.

5.5.2 NMT-mBART: Unseen Language Adaptation

As introduced in Section 5.4.2, the **NMT-mBART** model represents a pre-trained multilingual model fine-tuned using Quantized Low-Rank Adaptation combined with a two-phase curriculum learning schedule and back-translation data augmentation. Because this model was originally fine-tuned during a preliminary project, it utilized an unnormalized version of the training dataset. Despite this data quality disadvantage, the massive pre-trained cross-lingual parameter space of **NMT-mBART** significantly outperforms non-pretrained baselines. This architectural advantage is particularly evident in the Genoese-to-Italian direction (LIJ $\rightarrow$ ITA), where the model leverages strong pre-existing target representations of Italian to generate fluent output. Furthermore, for the reverse direction (ITA $\rightarrow$ LIJ), cross-lingual transfer from closely related Romance structures prevents sequence length degradation commonly observed in recurrent models.

Language-Specific Weight Transfer Analysis

To determine the optimal latent semantic representation for the newly added `<lij>` control token, three warm-start initialization strategies, cloning embeddings from Italian (`<it_IT>`), French (`<fr_XX>`), and Portuguese (`<pt_XX>`), were benchmarked on the in-domain custom test set. Table 5.8 summarizes the results across both translation directions.

Direction: Italian $\rightarrow$ Genoese (ITA $\rightarrow$ LIJ)			
Initialization Strategy	BLEU $\uparrow$	chrF++ $\uparrow$	TER $\downarrow$
Italian (<it_IT>)	31.38	59.23	46.26
French (<fr_XX>)	30.14	58.08	47.51
Portuguese (<pt_XX>)	31.89	59.15	49.15
Direction: Genoese $\rightarrow$ Italian (LIJ $\rightarrow$ ITA)			
Initialization Strategy	BLEU $\uparrow$	chrF++ $\uparrow$	TER $\downarrow$
Italian (<it_IT>)	53.05	72.87	30.99
French (<fr_XX>)	53.08	72.86	30.84
Portuguese (<pt_XX>)	52.98	72.81	31.11

Table 5.8: Study comparing control token (<li j>) embedding initialization strategies for NMT-mBART on the in-domain custom test set.

When translating into Genoese (ITA $\rightarrow$ LIJ), while Portuguese initialization achieves a marginally higher BLEU score (31.89), initializing the control token with Italian weights (<it_IT>) yields superior overall performance across morphological and structural metrics, attaining the top scores in both chrF++ (59.23) and TER (46.26). Conversely, when decoding into Italian (LIJ $\rightarrow$ ITA), all three initialization strategies perform almost identically, with French initialization (<fr_XX>) taking a narrow lead at 53.08 BLEU and 30.84 TER. This relative invariance indicates that when target outputs belong to a high-resource language natively present in NMT-mBART’s pre-training dataset, the decoder relies primarily on its pre-existing language model priors rather than the initial state of the source control token.

Full Multi-Benchmark Evaluation

Based on the ablation study, the most balanced configurations, Italian initialization for ITA $\rightarrow$ LIJ and French initialization for LIJ $\rightarrow$ ITA, were selected for comprehensive cross-benchmark evaluation. Table 5.9 details their performance across the Custom Test Set, Bouquet, and FLORES+.

NMT-mBART (Italian Init) Direction: Italian $\rightarrow$ Genoese (ITA $\rightarrow$ LIJ)			
Benchmark	BLEU $\uparrow$	chrF++ $\uparrow$	TER $\downarrow$
Custom Test Set	31.38	59.23	46.26
Bouquet	18.25	43.70	67.47
FLORES+	13.60	43.75	74.48

NMT-mBART (French Init) Direction: Genoese $\rightarrow$ Italian (LIJ $\rightarrow$ ITA)			
Benchmark	BLEU $\uparrow$	chrF++ $\uparrow$	TER $\downarrow$
Custom Test Set	53.08	72.86	30.84
Bouquet	30.72	56.79	57.68
FLORES+	20.73	51.58	66.51

Table 5.9: Quantitative performance of the best-performing NMT-mBART fine-tuned configurations across all evaluation benchmarks.

The cross-benchmark results demonstrate marked superiority over the custom NMT-BiLSTM baseline across all test environments. In the Genoese-to-Italian direction, NMT-mBART achieves 53.08 BLEU on the custom test split while maintaining strong out-of-domain generalization scores of 30.72 BLEU on Bouquet and 20.73 BLEU on FLORES+ (more than tripling the baseline’s 6.19 BLEU). Even in the more challenging ITA $\rightarrow$ LIJ direction, despite being trained on unnormalized source text, the model reaches 18.25 BLEU on Bouquet and 13.60 BLEU on FLORES+, more than doubling the generalization capacity of non-pretrained models. These findings confirm that massive pre-trained multilingual embeddings effectively cushion the impact of unnormalized training noise and mitigate out-of-domain performance degradation.

5.5.3 NMT-NLLB: Native Language Adaptation

Among the evaluated models, NMT-NLLB represents the most promising candidate due to its native multi-billion token pre-training by Meta, which explicitly encompasses Genoese (`lij_Latn`). To systematically quantify both the inherent multilingual capabilities of the model and the incremental gain provided by domain-specific adaptation, evaluation is conducted under two distinct settings across all three benchmarks: an initial off-the-shelf zero-shot baseline, followed by QLoRA fine-tuning (Section 5.2.1).

Fine-tuning experiments are further divided to assess the impact of training data composition, comparing model checkpoints trained strictly on parallel sentence pairs against those augmented with dictionary lexical pairs (Plus), synthetic backtranslated data (BT), and a combined setup (BT-Plus).

Off-the-Shelf Zero-Shot Evaluation

To establish an unbiased baseline, the unadapted model was evaluated directly on the three test corpora without any localized training or parameter updates. Table 5.10 details the zero-shot quantitative results for both translation directions.

Direction: Genoese $\rightarrow$ Italian (LIJ $\rightarrow$ ITA)			
Benchmark	BLEU $\uparrow$	chrF++ $\uparrow$	TER $\downarrow$
Custom Test Set	33.14	59.87	46.91
Bouquet	32.40	56.12	57.05
FLORES+	22.03	51.86	65.00
Direction: Italian $\rightarrow$ Genoese (ITA $\rightarrow$ LIJ)			
Benchmark	BLEU $\uparrow$	chrF++ $\uparrow$	TER $\downarrow$
Custom Test Set	11.59	39.81	78.35
Bouquet	10.70	34.71	84.37
FLORES+	9.62	38.12	82.92

Table 5.10: Quantitative performance of the off-the-shelf zero-shot NMT-NLLB baseline across evaluation benchmarks.

Similar to NMT-mBART, NMT-NLLB benefits significantly from its multi-billion token pre-training when translating into Italian, as decoding into a high-resource language leverages strong pre-existing linguistic priors. Across all three evaluation benchmarks, particularly on FLORES+, the off-the-shelf zero-shot NMT-NLLB model achieves performance highly comparable to fine-tuned NMT-mBART, outperforming it slightly in the Genoese-to-Italian direction (LIJ $\rightarrow$ ITA) on FLORES+ (22.03 vs. 20.73 BLEU) and Bouquet (32.40 vs. 30.72 BLEU).

Conversely, in the Italian-to-Genoese direction (ITA $\rightarrow$ LIJ), fine-tuned NMT-mBART significantly outperforms zero-shot NMT-NLLB across all benchmarks (e.g., 18.25 vs. 10.70 BLEU on Bouquet, and 13.60 vs. 9.62 BLEU on FLORES+).

Interestingly, on the Bouquet benchmark in the ITA $\rightarrow$ LIJ direction, even the non-pretrained NMT-BiLSTM baseline achieves a higher BLEU score (12.36) than zero-shot NMT-NLLB (10.70), while trailing by only a narrow margin in chrF++ (33.51 vs. 34.71). This disparity highlights the inherent limitations of the pre-training methodology employed for NMT-NLLB in ultra-low-resource scenarios. Meta’s native support for Genoese relied heavily on web-scraped data and Wikipedia text, corpora that are notably small and often inconsistent in orthography or translation quality. Consequently, while zero-shot pre-training provides sufficient semantic alignment to decode into a high-resource language like Italian, generating accurate, morphologically correct Genoese without localized, domain-specific fine-tuning remains severely constrained by the noise and limitations of its original pre-training data.

In-Domain Fine-Tuning Performance

To evaluate the impact of domain-specific adaptation, dictionary expansion, and synthetic data integration, QLoRA fine-tuning was evaluated across all variants in the Italian-to-Genoese direction. Table 5.11 presents the in-domain quantitative results on the custom test set.

Direction: Genoese $\rightarrow$ Italian (LIJ $\rightarrow$ ITA)			
Model Variant	BLEU $\uparrow$	chrF++ $\uparrow$	TER $\downarrow$
NMT-NLLB	58.77	76.24	28.07
NMT-NLLB-Plus (Dictionary Augmented)	59.13	76.56	27.67
Direction: Italian $\rightarrow$ Genoese (ITA $\rightarrow$ LIJ)			
Model Variant	BLEU $\uparrow$	chrF++ $\uparrow$	TER $\downarrow$
NMT-NLLB	55.11	73.74	29.66
NMT-NLLB-Plus (Dictionary Augmented)	56.18	74.47	28.98
NMT-NLLB-BT (Backtranslated Data)	55.06	73.89	29.62
NMT-NLLB-BT-Plus (Combined)	55.07	74.02	29.80

Table 5.11: In-domain performance comparison across NMT-NLLB fine-tuning variants on the custom test set.

The in-domain fine-tuning evaluation demonstrates that domain-specific adaptation yields a massive performance boost over the zero-shot baseline across both translation directions.

Incorporating dictionary lexical pairs consistently yields the top performance across all automated metrics, with **NMT-NLLB-Plus** achieving the highest overall score (56.18 BLEU and 74.47 chrF++ on ITA $\rightarrow$ LIJ), confirming that lexical replay helps consolidate vocabulary coverage without degrading sentence-level fluency.

Conversely, integrating synthetic backtranslated data (**NMT-NLLB-BT** and **NMT-NLLB-BT-Plus**) fails to provide a meaningful advantage, resulting in a slight performance plateau or minor drop in BLEU (55.06 and 55.07, respectively) compared to **NMT-NLLB-Plus** (56.18). Similar to the observations made for **NMT-EuroLLM** and **NMT-BiLSTM**, backtranslation does not yield substantial gains in this low-resource scenario. While backtranslation is designed to enhance target-side generation using monolingual corpora, the uncurated nature and orthographic inconsistency of the underlying monolingual source text (primarily derived from Wikivivagna) introduce noise that offsets any potential gain in fluency.

Full Multi-Benchmark Evaluation

To assess how domain-specific fine-tuning impacts out-of-domain generalization across distinct textual domains, the best-performing **NMT-NLLB-Plus** checkpoint was evaluated across all three test sets alongside the synthetic variants for completeness. Table 5.12 details the final cross-benchmark results.

NMT-NLLB-Plus Direction: Genoese $\rightarrow$ Italian (LIJ $\rightarrow$ ITA)			
Benchmark	BLEU $\uparrow$	chrF++ $\uparrow$	TER $\downarrow$
Custom Test Set	59.13	76.56	27.67
Bouquet	31.15	59.26	52.78
FLORES+	21.03	53.11	62.40
NMT-NLLB-Plus Direction: Italian $\rightarrow$ Genoese (ITA $\rightarrow$ LIJ)			
Benchmark	BLEU $\uparrow$	chrF++ $\uparrow$	TER $\downarrow$
Custom Test Set	56.18	74.47	28.98
Bouquet	26.94	52.11	57.09
FLORES+	17.24	48.95	69.41

Table 5.12: Quantitative performance of the best-performing dictionary-augmented **NMT-NLLB-Plus** model across all evaluation benchmarks.

The cross-benchmark evaluation of **NMT-NLLB-Plus** reveals distinct behavioral patterns depending on the translation direction. In the Italian-to-Genoese direction (ITA $\rightarrow$ LIJ), domain-specific fine-tuning yields a performance boost across all three evaluation environments compared to the zero-shot baseline. The most pronounced improvement occurs on the in-domain custom test set, where BLEU surges from 11.59 to 56.18. Out-of-domain generalization also improves significantly, with Bouquet increasing from 10.70 to 26.94 BLEU and FLORES+ rising from 9.62 to 17.24 BLEU (with **NMT-NLLB-BT-Plus** achieving a comparable 17.10 BLEU and slightly higher chrF++ of 49.03). As a result, **NMT-NLLB-Plus** establishes itself as a highly competitive model for translating into Genoese, outperforming fine-tuned **NMT-mBART** (18.25 Bouquet / 13.60 FLORES+) and the custom **NMT-BiLSTM** baseline.

Conversely, in the Genoese-to-Italian direction (LIJ $\rightarrow$ ITA), fine-tuning achieves a massive gain of nearly 26 BLEU points on the custom test set (59.13 vs. 33.14). However, on external benchmarks, fine-tuning induces a slight performance regression compared to the off-the-shelf zero-shot setup, with Bouquet dropping from 32.40 to 31.15 BLEU and FLORES+ decreasing from 22.03 to 21.03 BLEU. This subtle decline indicates a mild catastrophic forgetting effect, where adapting model parameters to the specific orthographic nuances of our Genoese dataset slightly compromises its native, highly generalized Italian generation capabilities on general-domain text.

5.5.4 **NMT-EuroLLM: Decoder-Only Generative Model**

Despite not being explicitly pre-trained on Genoese during its initial pre-training phase, **NMT-EuroLLM** represents a highly compelling candidate for low-resource translation. Modern instruction-tuned LLMs possess exceptional cross-lingual transfer and emergent generalization capabilities, making efficient parameter adaptation via QLoRA a state-of-the-art methodology for adapting multilingual foundation models to unseen regional languages.

In-Domain Data Composition Analysis

To evaluate the impact of different training data strategies, **NMT-EuroLLM** was fine-tuned across several data regimes: standard parallel sentence tuning, dictionary augmentation (**Plus**), backtranslation synthetic data integration (**BT**), and a combined setup (**BT-Plus**). Table 5.13 details the in-domain quantitative performance across all fine-tuned variants on the custom test split.

Direction: Genoese $\rightarrow$ Italian (LIJ $\rightarrow$ ITA)			
Model Variant	BLEU $\uparrow$	chrF++ $\uparrow$	TER $\downarrow$
NMT-EuroLLM Standard	69.78	82.27	20.81
NMT-EuroLLM-Plus (Dictionary Augmented)	69.90	82.60	20.61

Direction: Italian $\rightarrow$ Genoese (ITA $\rightarrow$ LIJ)			
Model Variant	BLEU $\uparrow$	chrF++ $\uparrow$	TER $\downarrow$
NMT-EuroLLM	67.55	80.77	21.94
NMT-EuroLLM-Plus (Dictionary Augmented)	67.41	80.72	22.24
NMT-EuroLLM-BT (Backtranslated Data)	65.24	79.22	24.02
NMT-EuroLLM-BT-Plus (Combined)	65.31	79.36	23.85

Table 5.13: In-domain performance comparison across NMT-EuroLLM fine-tuning variants on the custom test set.

NMT-EuroLLM establishes itself as the top-performing model overall on the in-domain custom dataset. In the Genoese-to-Italian direction (LIJ $\rightarrow$ ITA), the dictionary-augmented setup (NMT-EuroLLM-Plus) achieves the best performance across all three metrics, outperforming the standard variant by a narrow margin (69.90 BLEU and 82.60 chrF++). Conversely, in the Italian-to-Genoese direction (ITA $\rightarrow$ LIJ), standard fine-tuning on parallel sentences alone delivers the optimal performance (67.55 BLEU and 80.77 chrF++), slightly exceeding the dictionary-augmented baseline.

Similarly to the findings in the NMT-BiLSTM experiments, integrating synthetic backtranslated sentences degrades translation quality, causing a reduction of over 2 BLEU points and approximately 1.5 chrF++ points. However, given the high accuracy demonstrated by NMT-EuroLLM in the LIJ $\rightarrow$ ITA direction, this drop is unlikely to stem from poor translation quality during backtranslation. Instead, it is primarily attributable to the low initial quality and orthographic noise of the uncurated monolingual source text extracted from Wikivivagna.

Despite omitting backtranslation, NMT-EuroLLM achieves exceptional in-domain results with remarkable directional symmetry, showing only a minimal 2-point BLEU gap between decoding into Italian (69.90) and Genoese (67.55).

Full Multi-Benchmark Evaluation

To evaluate out-of-domain generalization across distinct textual domains, the optimal configurations for each translation direction, NMT-EuroLLM-Plus for LIJ $\rightarrow$ ITA and standard fine-tuned NMT-EuroLLM for ITA $\rightarrow$ LIJ, were benchmarked against the Bouquet and FLORES+ corpora. Table 5.14 summarizes the comprehensive multi-benchmark performance.

NMT-EuroLLM-Plus Direction: Genoese $\rightarrow$ Italian (LIJ $\rightarrow$ ITA)			
Benchmark	BLEU $\uparrow$	chrF++ $\uparrow$	TER $\downarrow$
Custom Test Set	69.90	82.60	20.61
Bouquet	32.92	61.09	50.78
FLORES+	22.43	54.59	60.01
NMT-EuroLLM Direction: Italian $\rightarrow$ Genoese (ITA $\rightarrow$ LIJ)			
Benchmark	BLEU $\uparrow$	chrF++ $\uparrow$	TER $\downarrow$
Custom Test Set	67.55	80.77	21.94
Bouquet	28.38	53.30	54.80
FLORES+	16.85	48.56	68.55

Table 5.14: Quantitative performance of the optimal NMT-EuroLLM fine-tuned configurations across all evaluation benchmarks.

In the Genoese-to-Italian direction (LIJ $\rightarrow$ ITA), NMT-EuroLLM outperforms NMT-NLLB-Plus across all three benchmarks and automated metrics, reaching 69.90 BLEU on the custom test set, 32.92 BLEU on Bouquet, and 22.43 BLEU on FLORES+, thereby establishing itself as the superior model for translating from Genoese into Italian.

In the reverse direction (ITA $\rightarrow$ LIJ), NMT-EuroLLM yields the highest overall quality on both the custom test set (67.55 BLEU) and Bouquet (28.38 BLEU). On FLORES+, while NMT-NLLB-Plus achieves a slightly higher BLEU score (17.24 vs. 16.85), NMT-EuroLLM maintains superior morphological accuracy and lower edit distance, outperforming NMT-NLLB-Plus in both chrF++ (48.56 vs. 48.95, where higher is better) and TER (68.55 vs. 69.41, where lower is better).

Consequently, NMT-EuroLLM emerges as the most effective overall model across both translation directions.

These empirical findings highlight a fundamental trade-off in low-resource machine translation. While fine-tuning large foundation models on specialized corpora produces extraordinary results on in-domain datasets, achieving comparable generalization on out-of-domain, general-knowledge text remains a persistent challenge.

The steep performance gap between in-domain test splits and external benchmarks underscores the inherent difficulty of training robust, broadly generalizable translation systems when constrained by limited data availability.

5.5.5 Comparative Analysis

To provide a comprehensive and unified synthesis of the quantitative evaluations, this section thoroughly compares the top-performing variant of each representative translation architecture: NMT-BiLSTM, fine-tuned NMT-mBART, fine-tuned NMT-NLLB-Plus, and fine-tuned NMT-EuroLLM. This comparative analysis spans across all three distinct target evaluation benchmarks to capture both general-domain capabilities and domain-specific challenges.

Specifically, Tables 5.15, 5.16, and 5.19 summarize the detailed comparative performance on the Custom Test Set, Bouquet, and FLORES+ benchmarks, respectively. By evaluating these architectures across diverse data distributions, lexical domains, and resource levels, we aim to highlight the distinct strengths, failure modes, and overall generalization limits exhibited by each model when applied to low-resource and dialectal translation tasks.

In-Domain Evaluation (Custom Test Set)

Table 5.15 details the comparative performance on the primary in-domain custom test split.

Direction: Genoese $\rightarrow$ Italian (LIJ $\rightarrow$ ITA)			
Model	BLEU $\uparrow$	chrF++ $\uparrow$	TER $\downarrow$
NMT-BiLSTM	38.46	59.16	44.98
NMT-mBART (French Init)	53.08	72.86	30.84
NMT-NLLB-Plus	59.13	76.56	27.67
NMT-EuroLLM-Plus	69.90	82.60	20.61

Direction: Italian $\rightarrow$ Genoese (ITA $\rightarrow$ LIJ)			
Model	BLEU $\uparrow$	chrF++ $\uparrow$	TER $\downarrow$
NMT-BiLSTM	36.46	55.28	48.21
NMT-mBART (Italian Init)	31.38	59.23	46.26
NMT-NLLB-Plus	56.18	74.47	28.98
NMT-EuroLLM	67.55	80.77	21.94

Table 5.15: Overall performance comparison of best-in-class models on the in-domain Custom Test Set.

On the in-domain test set, performance strictly correlates with scale and pre-training methodology. As illustrated in Figure 5.1, which depicts the BLEU scores achieved by the top-performing variant of each architecture across both translation directions, NMT-EuroLLM achieves absolute dominance in both settings, outperforming NMT-NLLB-Plus by over 10 BLEU points in LIJ $\rightarrow$ ITA (69.90 vs. 59.13) and over 11 BLEU points in ITA $\rightarrow$ LIJ (67.55 vs. 56.18).

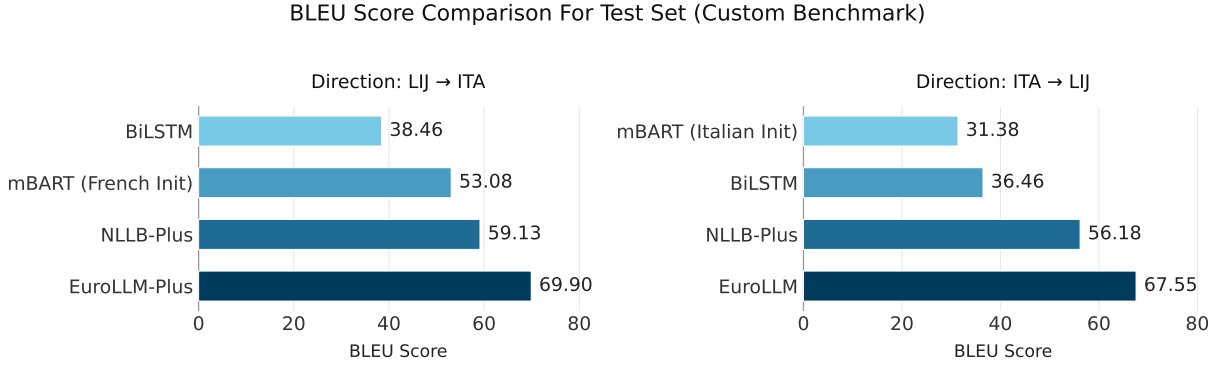


Figure 5.1: Comparative evaluation of BLEU scores across architectures on the in-domain Custom Test Set for both translation directions (LIJ → ITA and ITA → LIJ).

Literary Out-of-Domain Generalization (Bouquet Benchmark)

Table 5.16 presents the comparative evaluation on the Bouquet literary dataset.

Direction: Genoese → Italian (LIJ → ITA)			
Model	BLEU ↑	chrF++ ↑	TER ↓
NMT-BiLSTM	13.73	38.49	79.72
NMT-mBART (French Init)	30.72	56.79	57.68
NMT-NLLB-Plus	31.15	59.26	52.78
NMT-EuroLLM-Plus	32.92	61.09	50.78
Direction: Italian → Genoese (ITA → LIJ)			
Model	BLEU ↑	chrF++ ↑	TER ↓
NMT-BiLSTM	12.36	33.51	85.51
NMT-mBART (Italian Init)	18.25	43.70	67.47
NMT-NLLB-Plus	26.94	52.11	57.09
NMT-EuroLLM	28.38	53.30	54.80

Table 5.16: Overall performance comparison of best-in-class models on the Bouquet benchmark.

When tested on out-of-domain literary text from the Bouquet dataset (Figure 5.2), NMT-EuroLLM maintains its leadership in both directions, demonstrating superior robustness when translating complex literary structures into Genoese (28.38 BLEU vs. 26.94 for NMT-NLLB-Plus).

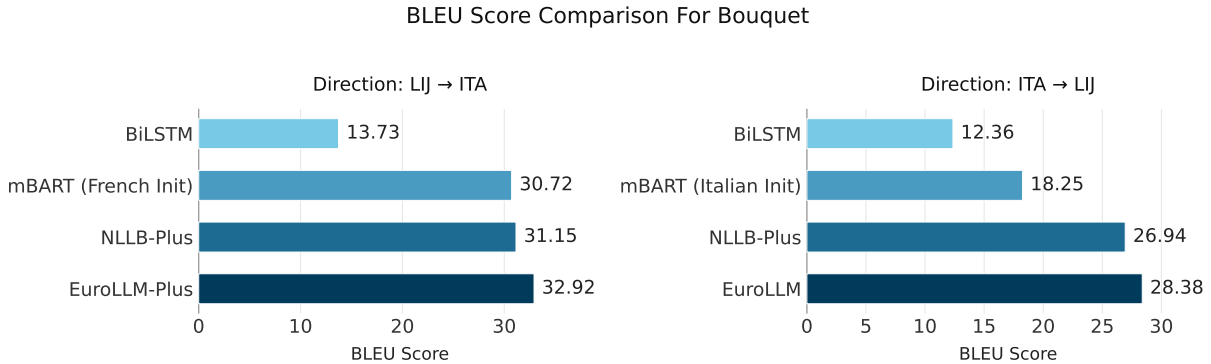


Figure 5.2: Comparative evaluation of BLEU scores across architectures on the out-of-domain Bouquet benchmark for both translation directions (LIJ → ITA and ITA → LIJ).

Following the evaluation, the best-performing models were successfully uploaded to Hugging Face, establishing their official positioning within the global Bouquet benchmark leaderboard across both translation directions, as detailed in Tables 5.17 and 5.18. It is important to highlight that the evaluation metrics reported in these tables were computed using the official Bouquet benchmark script available via Hugging Face. This implementation detail accounts for minor numerical discrepancies in the chrF++ results, compared to our local evaluations, specifically, while our local pipeline computes corpus-level scores via sentence-level averaging, the official evaluation script concatenates the entire corpus text prior to metric calculation.

Direction: Genoese → Italian (LIJ → ITA) - Hugging Face Leaderboard				
Model	chrF++ ↑	MetricX ↓	Glottid_score ↑	Glottid_x_metric ↑
OMT-NLLB	60.37	2.50	0.980	0.70
OMT-LLama-8B	58.72	2.58	0.982	0.69
nllb-200-3.3B	58.10	3.42	0.983	0.65
NMT-EuroLLM-Plus	56.90	2.45	0.979	0.69

Table 5.17: Official Hugging Face leaderboard evaluation metrics on the BOUQUET benchmark for Genoese → Italian, ordered by chrF++.

For the Genoese-to-Italian translation direction (Table 5.17), while NMT-EuroLLM-Plus ranks fifth when sorted strictly by chrF++, it notably achieves the **State-of-the-Art performance on MetricX** (6.95). This demonstrates superior semantic error minimization compared to top-ranking lexical models.

Furthermore, although massive architectures such as **Llama-3.3-70B-Instruct** achieve a peak GlotLID score reaching 0.983, slightly higher than the four models in the table, they suffer from a lower chrF++ score (53.49) and fail to secure top leaderboard placements. Despite lagging less than 4 points behind the absolute peak in chrF++, NMT-EuroLLM-Plus delivers remarkably robust overall performance across semantic and language-compliance axes, proving that fine-tuned mid-scale architectures can effectively rival massive foundational models by prioritizing true translation fidelity over raw surface matching.

Direction: Italian → Genoese (ITA → LIJ) - Hugging Face Leaderboard				
Model	chrF++ ↑	MetricX ↓	Glotlid_score ↑	Glotlid_x_metric ↑
NMT-EuroLLM	50.80	7.76	0.94	0.42
OMT-LLama-8B	46.53	7.41	0.92	0.42
OMT-LLama-3B	45.10	7.40	0.90	0.41
OMT-NLLB	44.82	7.10	0.94	0.44

Table 5.18: Official Hugging Face leaderboard evaluation metrics on the BOUQuET benchmark for Italian → Genoese, ordered by chrF++.

As presented in Table 5.18, NMT-EuroLLM achieves **State-of-the-Art performance in terms of chrF++** (50.80) for the Italian-to-Genoese translation direction on the Bouquet benchmark, outperforming existing models while maintaining competitive results across error-based and semantic metrics like MetricX and adjusted metrics.

While the **nllb-200-1.3B** model reaches a marginally higher language identification score on GlotLID (0.95), it suffers from a significantly lower chrF++ score (37.39), reflecting strong language confidence coupled with poor lexical and morphological adequacy. Consequently, such models fall short of the top-tier performance threshold required for the leaderboard.

In contrast, NMT-EuroLLM successfully strikes the optimal balance between target-language compliance and structural translation fidelity, cementing its superior generalization capabilities.

Multilingual Generalization (FLORES+ Benchmark)

Table 5.19 reports the comparative results on the general-domain FLORES+ benchmark.

Direction: Genoese $\rightarrow$ Italian (LIJ $\rightarrow$ ITA)			
Model	BLEU $\uparrow$	chrF++ $\uparrow$	TER $\downarrow$
NMT-BiLSTM	6.19	31.53	87.63
NMT-mBART (French Init)	20.73	51.58	66.51
NMT-NLLB-Plus	21.03	53.11	62.40
NMT-EuroLLM-Plus	22.43	54.59	60.01
Direction: Italian $\rightarrow$ Genoese (ITA $\rightarrow$ LIJ)			
Model	BLEU $\uparrow$	chrF++ $\uparrow$	TER $\downarrow$
NMT-BiLSTM	6.42	28.11	91.14
NMT-mBART (Italian Init)	13.60	43.75	74.48
NMT-NLLB-Plus	17.24	48.95	69.41
NMT-EuroLLM	16.85	48.56	68.55

Table 5.19: Overall performance comparison of best-in-class models on the FLORES+ benchmark.

On the FLORES+ benchmark (Figure 5.3), NMT-EuroLLM remains the top model for LIJ $\rightarrow$ ITA (22.43 BLEU). In the ITA $\rightarrow$ LIJ direction, while NMT-NLLB-Plus edges out NMT-EuroLLM by 0.39 BLEU points (17.24 vs. 16.85), NMT-EuroLLM compensates by achieving superior chrF++ (48.56) and lower TER (68.55), confirming its overall effectiveness across diverse evaluation scenarios.

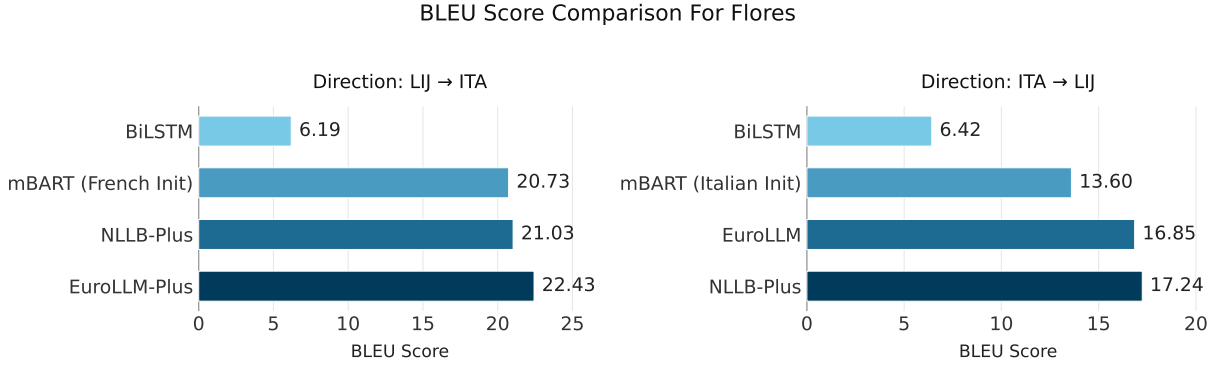


Figure 5.3: Comparative evaluation of BLEU scores across architectures on the out-of-domain FLORES+ benchmark for both translation directions (LIJ $\rightarrow$ ITA and ITA $\rightarrow$ LIJ).

5.5.6 Qualitative Analysis

To complement the quantitative evaluation, this section presents a qualitative error and performance breakdown of representative translation outputs extracted from the test set. Samples were selected by categorizing test sentences into three distinct performance groups based on their combined metric score (averaging sentence-level BLEU and chrF++): a high-score group, an intermediate-score group, and a low-score group. Predictions are evaluated using the optimal model configuration for each translation task, specifically NMT-EuroLLM-Plus for Genoese-to-Italian (LIJ $\rightarrow$ ITA) and standard fine-tuned NMT-EuroLLM for Italian-to-Genoese (ITA $\rightarrow$ LIJ).

Genoese to Italian Translation

The evaluation for the Genoese-to-Italian direction illustrates the model’s ability to navigate complex dialectal structures and translate them into fluent standard Italian using NMT-EuroLLM-Plus.

High-Score Performance Group	
Ex. 1	Source: o sabbo santo o l'é o giorno primma da pasqua
	Target: il sabato santo è il giorno prima della pasqua
	Pred: il sabato santo è il giorno prima della pasqua
Ex. 2	Source: a sciâ parödi a l'à promisso unn'attroveuia de çinquant'euro à chi ghe portià inderê o portafeuggio
	Target: la signora parodi ha promesso una ricompensa di cinquanta euro a chi le riporterà il portafoglio
	Pred: la signora parodi ha promesso una ricompensa di cinquanta euro a chi le riporterà il portafoglio
Ex. 3	Source: o figgeu o l'à beseugno de un scambello pe arrivâ a-o lavello
	Target: il bambino ha bisogno di uno sgabello per raggiungere il lavandino
	Pred: il bambino ha bisogno di uno sgabello per raggiungere il lavandino

Table 5.20: Qualitative examples extracted from the high-score performance group for Genoese to Italian.

Table 5.20 illustrates instances extracted from the high-score group, where predicted translations perfectly match the references. As discussed in previous sections, translating into standard Italian constitutes a less challenging task than the Italian-to-Genoese direction for pre-trained models, and specifically for a Large Language Model such as NMT-EuroLLM, due to the vast amount of Italian pre-training data available. Nevertheless, these examples demonstrate the model’s capacity to handle structural complexity in full-length sentences. Specifically, the model successfully resolves complex Genoese clitic systems and verbal constructions ‘*o l'é*’ → ‘*è*’, ‘*a l'à promisso*’ → ‘*ha promesso*’, maps specialized dialectal vocabulary ‘*attroveuia*’ → ‘*ricompensa*’, ‘*scambello*’ → ‘*sgabello*’, and correctly handles prepositional contractions ‘*da*’ → ‘*della*’, ‘*pe arrivâ a-o*’ → ‘*per raggiungere il*’ without introducing syntactic distortions.

Intermediate-Score Performance Group	
Ex. 4	Source: l'ötunno o l'é benvegnu, o cado o no me gusta!
	Target: ben venga l'autunno, il caldo non mi piace!
	Pred: l'autunno è benvenuto, il caldo non mi piace!
Ex. 5	Source: o l'é interessante nottâ che sta mecanica a l'é presente inti zeughi de tarocchi franseixi ascì.
	Target: è interessante notare che questa meccanica è presente anche nei giochi di tarocchi francesi.
	Pred: è interessante notare come questa meccanica sia presente anche nei giochi di tarocchi francesi.
Ex. 6	Source: taggiu de speck, salamme, crostoin euio, aggio, cornabuggia e tomata con da mossarella.
	Target: tagliere di speck, salame, crostoni olio, aglio, origano e pomodoro con mozzarella.
	Pred: tagliere di speck, salame, crostini all'olio, aglio, rosmarino e pomodoro con mozzarella.

Table 5.21: Qualitative examples extracted from the intermediate-score performance group for Genoese to Italian.

Table 5.21 presents predictions sampled from the intermediate-score group, where outputs diverge from the exact wording of the target reference. These examples illustrate a known limitation of automated evaluation metrics (e.g., BLEU, chrF, and TER) when applied to low-resource translation contexts, where reference datasets may contain paraphrasing or minor parallel alignment noise.

In Example 4, the model produces a literal and highly accurate translation of the source sentence ‘*l’autunno è benvenuto, il caldo non mi piace!*’, whereas the reference uses a more idiomatic phrasing ‘*ben venga l’autunno...*’. Despite generating an entirely valid translation, the model suffers an artificial metric penalty due to this reference discrepancy. Similarly, Example 5 exhibits a valid syntactic variation: rather than replicating the exact reference construction ‘*notare che questa meccanica è...*’, the model opts for a stylistic variant using a subjunctive clause ‘*notare come questa meccanica sia...*’. While semantically identical and grammatically pristine, this shift is penalized by n-gram matching algorithms.

Finally, Example 6 highlights minor lexical errors within an otherwise sound translation: the model misinterprets the Genoese term for oregano ‘*cornabuggia*’, predicting ‘*rosmarino*’ instead of ‘*origano*’, slightly alters the noun form ‘*crostini*’ instead of ‘*crostoni*’, and inserts an additional preposition ‘*all*’. Overall, these instances demonstrate that moderate metric drops often stem from stylistic divergences or minor lexical shifts rather than structural translation failures.

Low-Score Performance Group	
Ex. 7	Source: o fuxibile o s’é bruxou e l’é anæto via a luxe
	Target: il fusibile si è bruciato e è saltata la corrente
	Pred: il fusibile si è bruciato e si è spento la luce
Ex. 8	Source: l’obiettivo da reġerca a l’é unna neuva vacciña
	Target: l’oggetto della ricerca è un nuovo vaccino
	Pred: l’ obiettivo dello studio è una nuova vaccinazione
Ex. 9	Source: sciù, pöchi descorsci, dimme!
	Target: dai su, poche chiacchiere, dimmi!
	Pred: su , poche parole , dimmi!

Table 5.22: Qualitative examples extracted from the low-score performance group for Genoese to Italian.

Table 5.22 reports examples sampled from the low-score group, which exhibit lower metric scores due to more pronounced translation errors, literal transfers, and reference misalignments.

In Example 7, the model struggles with agreement and domain-specific idiomatic expressions: it predicts ‘*si è spento la luce*’, making a gender agreement error ‘*spento*’ instead of ‘*spenta*’ and failing to capture the target’s idiomatic rendering ‘*è saltata la corrente*’.

Interestingly, a direct translation of the source ‘*l’è anæto via a luxe*’ would yield ‘*è andata via la luce*’, revealing that the reference itself deviates from a literal transfer, even though the model’s generated output remains grammatically flawed. In Example 8, the model demonstrates superior fidelity to the source by correctly translating ‘*obiiettivo*’ rather than opting for the target’s looser rendering ‘*oggetto*’. However, it introduces a semantic hallucination at the sentence boundary, predicting ‘*vaccinazione*’ instead of the precise noun ‘*vaccino*’. Finally, Example 9 illustrates a failed transfer of pragmatic markers and colloquial expressions: the model translates the introductory particle ‘*sciù*’ literally as ‘*su*’ omitting the natural Italian discourse marker ‘*dai*’, and replaces ‘*descorsi*’ with the literal equivalent ‘*parole*’, failing to capture the idiomatic sense of the target expression ‘*chiacchiere*’ meaning ‘get to the point’.

In summary, low-score group outputs highlight that performance degradation in this direction is primarily driven by difficulty in mapping dialectal colloquialisms, localized grammatical agreement errors, and occasional reference discrepancies in the test set.

Italian to Genoese Translation

Translating into Genoese (ITA → LIJ) represents a substantially more challenging generation task than the reverse direction due to target-side morphological complexity, rich clitic agreement rules, and non-standardized orthographic conventions. Here, the standard fine-tuned NMT-EuroLLM model is evaluated.

High-Score Performance Group	
Ex. 1	Source: chi lavora mangia un’acciuga, chi non lavora ne mangia due.
	Target: chi loua mangia unn’ancioa, chi no loua o ne mangia doe.
	Pred: chi loua mangia unn’ancioa, chi no loua o ne mangia doe.
Ex. 2	Source: la francia riconoscerà lo stato di palestina
	Target: a fransa a reconoscià o stato de palestiña
	Pred: a fransa a reconoscià o stato de palestiña
Ex. 3	Source: il mazziere prende tre carte e le volta
	Target: o massê o piggia træ carte e o ê gia
	Pred: o massê o piggia træ carte e o ê gia

Table 5.23: Qualitative examples extracted from the high-score performance group for Italian to Genoese.

Table 5.23 presents top-quantile predictions where the model produces flawless outputs perfectly aligning with the reference translations. In Example 1, the model captures the proverbial sentence structure, correctly predicting Genoese verb forms ‘*lavora*’ → ‘*loua*’, article elision ‘*unn’anciaa*’, subject clitic pronouns ‘*o ne mangia*’, and feminine numeral inflections ‘*due*’ → ‘*doe*’. Example 2 illustrates accurate diplomatic news translation, handling subject clitic doubling ‘*a fransa a...*’, future tense morphology ‘*reconoscià*’, and standardized Genoese diacritics ‘*palestiña*’. In Example 3, the model correctly translates specialized card-game vocabulary ‘*mazziere*’ → ‘*massê*’, handles plural feminine numerals ‘*tre*’ → ‘*træ*’, and successfully predicts complex Genoese object-clitic contractions ‘*e le volta*’ → ‘*e o ê gia*’.

Intermediate-Score Performance Group	
Ex. 4	Source: il 16 ottobre le precipitazioni hanno colpito la provincia di savona.
	Target: a-i 16 d’ottobre e preçipitaçioin an bersaggiou a provinsa de saña.
	Pred: a-i 16 d’ottobre e preçipitaçioin an corpio a provinsa de saña.
Ex. 5	Source: ma no, su, dov’è tutta questa complessità?
	Target: ma no, sciù, che pöi dond’a l’é tutta sta complessità?
	Pred: ma no, sciù, dond’a l’é tutta sta complessità?
Ex. 6	Source: forza genoa!
	Target: dagghe, zena!
	Pred: alê zena!

Table 5.24: Qualitative examples extracted from the intermediate-score performance group for Italian to Genoese.

Table 5.24 illustrates intermediate-score group predictions where generated outputs present semantically equivalent or valid alternative renderings that are penalized by automated reference matching. In Example 4, the model translates ‘*hanno colpito*’ literally as ‘*an corpio*’, whereas the target reference employs a more expressive, domain-specific metaphor ‘*an bersaggiou*’. While the prediction remains grammatically flawless and semantically precise, it incurs an artificial metric penalty. In Example 5, the model produces a concise, accurate translation of the source discourse marker ‘*dond’a l’é tutta sta complessità?*’, whereas the reference contains an optional discourse filler ‘*che pöi*’.

Finally, Example 6 presents a valid stylistic choice: when translating the popular football chant ‘*forza genoa!*’, the model generates the native cheer ‘*alê zena!*’, whereas the reference split selects the imperative expression ‘*dagghe, zena!*’. Both renderings represent completely authentic Genoese expressions, underscoring how rigid evaluation metrics often underestimate true generation quality in idiomatic and colloquial settings.

Low-Score Performance Group	
Ex. 7	Source: datemi un minuto per considerare la situazione
	Target: dæme un segundo pe examinâ a scituacion
	Pred: dæme un menuto pe raxonâ da scituacion
Ex. 8	Source: a pancia piena si parla meglio
	Target: co-a pansa piña se raxoña ciù ben
	Pred: à pansa piña se parla megio
Ex. 9	Source: il salone principale è spazioso e ben illuminato
	Target: o sallon primmäio o l'é ampio e ben ciaeo
	Pred: o sallon prinçipâ o l'é spaçioso e ben inluminou

Table 5.25: Qualitative examples extracted from the low-score performance group for Italian to Genoese.

Table 5.25 reports low-score group predictions, which exhibit structural defects, literal calques, and reference misalignments. In Example 7, while the model accurately translates ‘*minuto*’ as ‘*menuto*’, unlike the target which loosely translates it as ‘*segondo*’, it introduces a verb construction error by predicting ‘*raxonâ da*’ instead of the transitive target ‘*examinâ*’. Example 8 displays a literal transfer of Italian prepositional phrases and vocabulary: instead of adopting the idiomatic Genoese construction ‘*co-a pansa piña se raxoña ciù ben*’, the model calques the Italian structure into ‘*à pansa piña se parla megio*’, failing to select the authentic dialectal verb ‘*raxonâ*’.

Finally, Example 9 provides a compelling case where the model’s generated hypothesis proves superior in fidelity and precision to the ground-truth target. While the reference opts for a freer, non-literal translation ‘*primmäio*’, ‘*ampio*’, ‘*ciaeo*’, the model predicts ‘*o sallon prinçipâ o l'é spaçioso e ben inluminou*’. Here, the model correctly preserves the exact semantic nuances of the source text, accurately rendering ‘*principale*’ as ‘*prinçipâ*’, ‘*spazioso*’ as ‘*spaçioso*’, and ‘*illuminato*’ as ‘*inluminou*’, while seamlessly applying correct Genoese phonotactics, orthographic normalization, and subject clitic agreement ‘*o... o l'é*’.

In conclusion, performance degradation in the low-score group for Italian to Genoese is driven either by cross-lingual interference, where the model defaults to literal Italian syntactic structures rather than distinct dialectal idioms, or by reference discrepancies where the model’s accurate, faithful output is artificially penalized by freer ground-truth translations.

Chapter 6

Conclusions and Future Work

This thesis addressed the challenge of neural machine translation for the Genoese dialect of Ligurian, evaluating translation both from Genoese into Italian and vice versa. The primary obstacles inherent to this task stem from the severe scarcity of parallel data and the absence of a universally adopted orthographic standard.

6.1 Summary of Contributions and Main Findings

To mitigate orthographic inconsistency, a significant portion of this work focused on developing a dedicated text normalization pipeline. Progressing from rule-based symbolic approaches to recurrent neural models, we devised a normalization sequence designed to harmonize Genoese orthographic variants, thereby simplifying the cross-lingual mapping for downstream translation architectures. The proposed attention-augmented recurrent normalizer achieved strong performance on the in-domain dataset, reducing the Character Error Rate (CER) down to 1.33% on sentence-level test sets and successfully outperforming the previous state-of-the-art baseline of 2.47% by over one percentage point.

Text normalization proved crucial in preventing the translation models from encountering redundant spelling variations during training, thereby reducing structural ambiguity. Furthermore, this preprocessing step helped mitigate the known brittleness of standard automated evaluation metrics, such as Bilingual Evaluation Understudy (BLEU), Translation Edit Rate (TER), and Character n-gram F-score (chrF++), which heavily penalize minor character discrepancies and valid lexical alternatives in low-resource settings. As confirmed by our qualitative analysis, many outputs categorized in low-score performance groups were in fact grammatically sound and semantically accurate, often producing valid synonyms or more natural phrasing than the ground-truth references.

In the machine translation phase, our comparative evaluation demonstrated that leveraging massive foundational models significantly outperforms lightweight recurrent architectures trained from scratch. Specifically, the generative causal language model, **NMT-EuroLLM**, emerged as the top-performing architecture across both translation directions. By fine-tuning this model on our normalized dataset, we effectively overcame the low-resource bottleneck, achieving exceptional fluency and semantic fidelity even on complex dialectal structures.

To further validate these findings beyond local evaluations, our top-performing models, **NMT-EuroLLM** and **NMT-EuroLLM-Plus**, were evaluated using the official BOUQuET benchmark repository and subsequently submitted to the Hugging Face leaderboard. Across the standardized evaluation metrics, the two **NMT-EuroLLM** models successfully secured competitive leaderboard standings and State-of-the-Art achievements in two different metrics in both directions: chrF++ for Italian-to-Genoese and MetricX for Genoese-to-Italian. This external validation confirms that fine-tuning foundational architectures on curated regional datasets effectively bridges the gap in low-resource machine translation, offering a scalable and robust alternative to massive general-purpose models.

Throughout the experimental phase, data augmentation strategies yielded mixed results. Synthetic data expansion proved effective primarily for the normalization task through back-normalization, whereas rule-based data augmentation derived from symbolic techniques failed to yield measurable gains.

Surprisingly, back-translation did not improve performance across any of the tested translation models, leading instead to a performance degradation. Several factors account for this outcome. For smaller architectures like **NMT-BiLSTM**, the degradation is attributed to the low initial translation quality in the LIJ $\rightarrow$ ITA direction, which generated noisy synthetic Italian targets, adding a large volume of such low-quality parallel pairs likely saturated the model capacity, triggering catastrophic forgetting.

For larger pre-trained models, additional domain-specific bottlenecks emerged. First, the unaligned Genoese source data harvested from web sources (*Wikivivagna*) lacked strict editorial control or adherence to standardized orthographies such as *grafia oficiâ* or *grafia unitäia*, consequently, severely non-standard instances remained noisy even after passing through the text normalization pipeline. Second, fine-tuning via parameter-efficient fine-tuning methods, specifically QLoRA, updates only a small fraction of trainable parameters. Introducing a synthetic corpus significantly larger than the original authentic parallel dataset likely saturated the capacity of these trainable adapters. Fully fine-tuning a larger set of parameters to accommodate the expanded synthetic data would have required computational resources beyond the scope of this project, and was deemed unnecessary given the competitive performance achieved by the models without back-translation.

6.2 Future Directions

To further advance performance and address the primary bottleneck of data scarcity in Genoese, several promising avenues emerge for future research. Extending the authentic parallel dataset by harvesting high-quality alignments from domain sources unrepresented in this work would broaden the training domain, thereby improving out-of-domain robustness on benchmarks such as FLORES+ and Bouquet.

Regarding synthetic data generation, refined data filtering pipelines could be applied to large monolingual Genoese repositories, such as the 160000-sentence corpus leveraged in this study, to filter out spelling inconsistencies and extract high-quality subsets for back-translation. Additionally, targeted vocabulary augmentation strategies, such as the contextual substitution of frequent terms with rare lexical items [18], warrant systematic exploration. While omitted in this study due to the intensive manual validation required to prevent semantic degradation, implementing automated quality-estimation filters could make contextual substitution a viable pathway to enhance the translation of low-frequency Genoese terms.

Furthermore, addressing the inherent limitations of automated evaluation represents another critical direction for future work. As previously highlighted, standard reference-based metrics such as BLEU, TER, and chrF++ struggle to accurately assess translation quality in low-resource settings. A promising alternative involves deploying LLM-based evaluation frameworks guided by structured rubric protocols, which have demonstrated significant potential in accurately evaluating low-resource languages (e.g., Zouhar et al. [74]). Through carefully engineered prompts and evaluation rubrics, large language models can perform semantic assessment and assign numerical quality scores while accounting for valid lexical variations, synonyms, and idiomatic expressions that traditional metrics penalize. While exploiting domain experts through human evaluation remains the gold standard for low-resource language assessment, LLM-based evaluation offers a scalable complementary tool to bridge the gap between automated scoring and human judgment.

Appendix A

Experimental Setup and Hardware Infrastructure

This appendix provides technical specifications for the hardware, operating systems, and computing infrastructure utilized across all experiments in this thesis, encompassing both orthographic normalization (Chapter 4) and machine translation (Chapter 5).

Local Workstation Specifications

The baseline experiments, deterministic data perturbations, and less resource-intensive training pipelines for the normalization task, specifically the rule-based approach (Section 4.3.2) and the Conditional Random Fields optimization grid (Section 4.3.3), were conducted locally on a personal workstation.

This local environment featured an AMD Ryzen 7 PRO 7840U processor operating at a base frequency of 3.3 GHz, configured with 8 physical cores and 16 logical threads to enable efficient parallel CPU processing. The workstation was equipped with 32 GB of physical RAM and operated under the Microsoft Windows 11 Home operating system (Build 26200). Due to the architectural constraints of the integrated AMD Radeon 780M graphics chip, these local configurations relied exclusively on multi-threaded CPU execution without dedicated hardware GPU acceleration.

High-Performance Computing Infrastructure

To handle the computationally intensive training, fine-tuning, and data augmentation phases of the complex deep learning architectures across both chapters, experiments were scaled up to high-performance computing infrastructures.

For the general architecture experiments, the dedicated server node was powered by an Intel Xeon Silver 4410Y processor alongside 512 GB of system RAM, running under the Ubuntu 24.04.4 LTS operating system. Accelerated deep learning execution within this remote environment was performed on an enterprise-grade NVIDIA L40S GPU built on the Ada Lovelace architecture, featuring 46 GB of dedicated VRAM and operating under the CUDA 13.0 computational framework. The associated software stack comprised Python 3.12 and PyTorch 2.12.1 with CUDA 13.0 support, utilizing Hugging Face Transformers v5.14.1 for model management, Datasets v5.0.0 and TRL v1.9.1 for training workflows, PEFT v0.19.1 and BitsAndBytes v0.50.0 for efficient adaptation, Evaluate v0.4.6 for metric computation, and SacreBLEU v2.6.0 to compute standardized BLEU and chrF++ scores.

Conversely, the training and fine-tuning phases specifically dedicated to the mBART model were carried out on a separate dedicated workstation. This setup relied on an 11th Gen Intel Core i9-11900KF processor operating at 3.50 GHz with 8 cores and 16 threads, accompanied by 32 GB of DDR4 RAM. Accelerated execution was handled by an NVIDIA GeForce RTX 3060 GPU with 12 GB of VRAM, running on Driver 555.42 with CUDA 12.5 support. The software stack comprised Python 3.12 and PyTorch 2.5.1 with CUDA 12.1 support, utilizing Hugging Face Transformers v5.4.0 for model management, Evaluate v0.4.6 for metric computation, and SacreBLEU v2.6.0 to compute standardized BLEU and chrF++ scores.

Bibliography

- [1] Académia Ligùstica do Brénno. Académia ligùstica do brénno. URL: <https://www.zeneize.net/index.html>.
- [2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL: <http://arxiv.org/abs/1409.0473>.
- [3] Franco Bampi. Traduttore italiano genovese - TIG. URL: <https://www.zeneize.net/itze/main.asp>.
- [4] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, 2003. URL: <https://jmlr.org/papers/v3/bengio03a.html>.
- [5] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL: <https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfc4967418bfb8ac142f64a-Abstract.html>.
- [6] Laurie Burchell, Jean Maillard, Antonios Anastasopoulos, Christian Federmann, Philipp Koehn, and Skyler Wang. Findings of the WMT 2024 shared task of the open language data initiative. In Barry Haddow, Tom Kocmi, Philipp Koehn, and

- Christof Monz, editors, *Proceedings of the Ninth Conference on Machine Translation*, pages 110–117, Miami, Florida, USA, November 2024. Association for Computational Linguistics. URL: <https://aclanthology.org/2024.wmt-1.4/>, doi:10.18653/v1/2024.wmt-1.4.
- [7] Davide Buscaldi and Paolo Rosso. How good is NLLB for low-resource languages? A study on the genoese language. In Federico Boschetti, Gianluca E. Leboni, Bernardo Magnini, and Nicole Novielli, editors, *Proceedings of the 9th Italian Conference on Computational Linguistics, Venice, Italy, November 30 - December 2, 2023*, volume 3596 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2023. URL: <https://ceur-ws.org/Vol-3596/short5.pdf>.
 - [8] Giovanni Casaccia. *Dizionario genovese-italiano*. Franchielli, Genoa, Italy, 2nd edition, 1876. Seconda edizione accresciuta del doppio e quasi tutta rifatta.
 - [9] Sara Caviglia. Parts of speech: the genoese dialect, 2025. Natural Language Processing project, University of Genoa. URL: <https://github.com/saracaviglia/NLP-project.git>.
 - [10] Ján Cegin, Daniil Gurgurov, Yusser Al Ghussin, and Simon Ostermann. Want better synthetic data? steer it: Activation steering for low-resource language generation, 2026. URL: <https://doi.org/10.48550/arXiv.2606.18389>, arXiv: 2606.18389, doi:10.48550/ARXIV.2606.18389.
 - [11] Peter J. A. Cock, Tiago Rodrigues Antao, Jeffrey T. Chang, Brad A. Chapman, Cymon J. Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, and Michiel J. L. de Hoon. Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinform.*, 25(11):1422–1423, 2009. URL: <https://doi.org/10.1093/bioinformatics/btp163>, doi: 10.1093/BIOINFORMATICS/BTP163.
 - [12] Conseggio pe-o patrimònio linguistico ligure. Conseggio pe-o patrimònio linguistico ligure. URL: <https://conseggio-ligure.org/en>.
 - [13] Conseggio pe-o patrimònio linguistico ligure. Dicionário eletrônico italian-zeneise (deize). URL: <https://conseggio-ligure.org/it/dizionario/deize>.
 - [14] Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Y. Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loïc Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj

- Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. No language left behind: Scaling human-centered machine translation. *CoRR*, abs/2207.04672, 2022. URL: <https://doi.org/10.48550/arXiv.2207.04672>, arXiv:2207.04672, doi:10.48550/ARXIV.2207.04672.
- [15] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL: http://papers.nips.cc/paper_files/paper/2023/hash/1feb87871436031bdc0f2beaa62a049b-Abstract-Conference.html.
- [16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL: <http://arxiv.org/abs/1810.04805>, arXiv:1810.04805.
- [17] Sardo Fabrizio. Zenamas, 2025. Natural Language Processing Symbolic and Distributed Artificial Intelligence project, University of Genoa. URL: <https://github.com/fabriziosardo/ZenaMAS.git>.
- [18] Marzieh Fadaee, Arianna Bisazza, and Christof Monz. Data augmentation for low-resource neural machine translation. volume abs/1705.00440, 2017. URL: <http://arxiv.org/abs/1705.00440>, arXiv:1705.00440.
- [19] Angelico Federico Gazzo. *A Diviña comédia de Dante di Ardighê tradûta in léngua zeneyze cu 'i segni da pronúncia*. Stampaya da Zonent..., Genoa, Italy, 1909.
- [20] Christopher R. Haberland, Jean Maillard, and Stefano Lusito. Italian-ligurian machine translation in its cultural context. In Maite Melero, Sakriani Sakti, and Claudia Soria, editors, *Proceedings of the 3rd Annual Meeting of the Special Interest Group on Under-resourced Languages @ LREC-COLING 2024*, pages 168–176, Turin, Italy, May 2024. European Language Resources Association (ELRA) and ICCL. doi:10.63317/28wcf56ofii2.
- [21] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997. URL: <https://doi.org/10.1162/neco.1997.9.8.1735>, doi:10.1162/NECO.1997.9.8.1735.
- [22] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th*

- Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 328–339. Association for Computational Linguistics, 2018. URL: <https://aclanthology.org/P18-1031/>, doi:10.18653/V1/P18-1031.
- [23] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685, 2021. URL: <https://arxiv.org/abs/2106.09685>, arXiv:2106.09685.
- [24] Il Secolo XIX. Il secolo xix. URL: <https://www.ilsecoloxix.it/>.
- [25] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. Mistral 7b. *CoRR*, abs/2310.06825, 2023. URL: <https://doi.org/10.48550/arXiv.2310.06825>, arXiv:2310.06825, doi:10.48550/ARXIV.2310.06825.
- [26] Sethunya Joseph, Kutlwano Sedimo, F. Kaniwa, Hlomani Hlomani, and Keletso Letsholo. Natural language processing: A review. *Natural Language Processing: A Review*, 6:207–210, 03 2016.
- [27] Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, Andr   F. T. Martins, and Alexandra Birch. Marian: Fast neural machine translation in C++. In Fei Liu and Thamar Solorio, editors, *Proceedings of ACL 2018, Melbourne, Australia, July 15-20, 2018, System Demonstrations*, pages 116–121. Association for Computational Linguistics, 2018. URL: <https://aclanthology.org/P18-4020/>, doi:10.18653/V1/P18-4020.
- [28] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*. 3rd edition, 2026. Online manuscript released August 19, 2026. URL: <https://web.stanford.edu/~jurafsky/slp3/>.
- [29] Juraj Juraska, Daniel Deutsch, Mara Finkelstein, and Markus Freitag. Metricx-24: The google submission to the wmt 2024 metrics shared task, 2024. URL: <https://arxiv.org/abs/2410.03983>, arXiv:2410.03983.
- [30] Amir Kargaran, Ayyoob Imani, Fran  ois Yvon, and Hinrich Sch  etze. Glotlid: Language identification for low-resource languages. In *Findings of the Association for*

- Computational Linguistics: EMNLP 2023*, page 6155–6218. Association for Computational Linguistics, 2023. URL: <http://dx.doi.org/10.18653/v1/2023.findings-emnlp.410>, doi:10.18653/v1/2023.findings-emnlp.410.
- [31] Philipp Koehn. *Neural Machine Translation*. Cambridge University Press, 2020.
- [32] Zhiyuan Liu, Yankai Lin, and Maosong Sun, editors. *Representation Learning for Natural Language Processing*. Springer Singapore, 2nd edition, 2023. doi:10.1007/978-981-99-1600-9.
- [33] Lunäio zeneise da sciâ Tägnâ. Lunäio zeneise da sciâ tägnâ. URL: <https://lunaio.org/>.
- [34] Marian Lupascu, Ana-Cristina Rogoz, Mihai-Sorin Stupariu, and Radu Tudor Ionescu. Large multimodal models for low-resource languages: A survey. *Inf. Fusion*, 131:104189, 2026. URL: <https://doi.org/10.1016/j.inffus.2026.104189>, doi:10.1016/J.INFFUS.2026.104189.
- [35] Massimo Lusetti, Tatyana Ruzsics, Anne Göhring, Tanja Samardzic, and Elisabeth Stark. Encoder-decoder methods for text normalization. In Marcos Zampieri, Preslav Nakov, Nikola Ljubesic, Jörg Tiedemann, Shervin Malmasi, and Ahmed Ali, editors, *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects, VarDial@COLING 2018, Santa Fe, New Mexico, USA, August 20, 2018*, pages 18–28. Association for Computational Linguistics, 2018. URL: <https://aclanthology.org/W18-3902/>.
- [36] Stefano Lusito, Edoardo Ferrante, and Jean Maillard. Text normalization for low-resource languages: the case of ligurian. In *Proceedings of the Sixth Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 98–103, 2023.
- [37] Courtney Mansfield, Ming Sun, Yuzong Liu, Ankur Gandhe, and Björn Hoffmeister. Neural text normalization with subword units. In Anastassia Loukina, Michelle Morales, and Rohit Kumar, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 2 (Industry Papers)*, pages 190–196. Association for Computational Linguistics, 2019. URL: <https://doi.org/10.18653/v1/n19-2024>, doi:10.18653/V1/N19-2024.
- [38] Pedro Henrique Martins, Patrick Fernandes, João Alves, Nuno Miguel Guerreiro, Ricardo Rei, Duarte M. Alves, José Pombal, M. Amin Farajian, Manuel Faysse, Mateusz Klimaszewski, Pierre Colombo, Barry Haddow, José G. C. de Souza, Alexandra

- Birch, and André F. T. Martins. Eurollm: Multilingual language models for europe. *CoRR*, abs/2409.16235, 2024. URL: <https://doi.org/10.48550/arXiv.2409.16235>, arXiv:2409.16235, doi:10.48550/ARXIV.2409.16235.
- [39] Dietram Efrem Mgeni. Employing large language models in swahili, a low-resource language. *Nat. Lang. Process. J.*, 15:100209, 2026. URL: <https://doi.org/10.1016/j.nlp.2026.100209>, doi:10.1016/J.NLP.2026.100209.
- [40] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013. URL: <http://arxiv.org/abs/1301.3781>.
- [41] Christopher Moseley. *Atlas of the World’s Languages in Danger*. Memory of Peoples. UNESCO Publishing, Paris, 3rd edition, 2010.
- [42] Pavan Naik, Ankit Jaiswal, Inacio Vieira, and Anya Belz. Improving low-resource language translation via targeted vocabulary expansion. In *Irish Conference on Artificial Intelligence and Cognitive Science*, pages 336–347. Springer, 2026.
- [43] Saul B. Needleman and Christian D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, 1970. URL: <https://www.sciencedirect.com/science/article/pii/0022283670900574>, doi:10.1016/0022-2836(70)90057-4.
- [44] O Balilla. O balilla, 1868–1904.
- [45] O Zinâ. O zinâ. Project initiated by Conseggio pe-o patrimònio linguistico ligure, edited by Fabio Canessa. URL: <https://ozina.org/>.
- [46] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL: http://papers.nips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html.
- [47] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin, editors, *Proceedings of the 40th Annual Meeting of the Association*

- for *Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. URL: <https://aclanthology.org/P02-1040/>, doi:10.3115/1073083.1073135.
- [48] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL, 2014. URL: <https://doi.org/10.3115/v1/d14-1162>, doi:10.3115/V1/D14-1162.
 - [49] Martin Piaggio. *Raccolta delle migliori poesie edite e inedite*. Fratelli Pagano, Genoa, Italy, 1846.
 - [50] Maja Popović. chrF: character n-gram F-score for automatic MT evaluation. In Ondřej Bojar, Rajan Chatterjee, Christian Federmann, Barry Haddow, Chris Hokamp, Matthias Huck, Varvara Logacheva, and Pavel Pecina, editors, *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal, September 2015. Association for Computational Linguistics. URL: <https://aclanthology.org/W15-3049/>, doi:10.18653/v1/W15-3049.
 - [51] Maja Popović. chrF++: words helping character n-grams. In Ondřej Bojar, Christian Buck, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, and Julia Kreutzer, editors, *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. URL: <https://aclanthology.org/W17-4770/>, doi:10.18653/v1/W17-4770.
 - [52] Simon J.D. Prince. *Understanding Deep Learning*. The MIT Press, 2023. URL: <http://udlbook.com>.
 - [53] Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. In *Improving Language Understanding by Generative Pre-Training*, 2018. URL: <https://api.semanticscholar.org/CorpusID:49313245>.
 - [54] Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016. URL: <https://doi.org/10.18653/v1/p16-1009>, doi:10.18653/V1/P16-1009.
 - [55] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting*

- of the Association for Computational Linguistics, *ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016. URL: <https://doi.org/10.18653/v1/p16-1162>, doi:10.18653/V1/P16-1162.
- [56] Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA, August 8-12 2006. Association for Machine Translation in the Americas. URL: <https://aclanthology.org/2006.amta-papers.25/>.
- [57] Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. Curriculum learning: A survey. *Int. J. Comput. Vis.*, 130(6):1526–1565, 2022. URL: <https://doi.org/10.1007/s11263-022-01611-x>, doi:10.1007/S11263-022-01611-X.
- [58] Richard Sproat and Navdeep Jaitly. An RNN model of text normalization. In Francisco Lacerda, editor, *18th Annual Conference of the International Speech Communication Association, Interspeech 2017, Stockholm, Sweden, August 20-24, 2017*, pages 754–758. ISCA, 2017. URL: <https://doi.org/10.21437/Interspeech.2017-35>, doi:10.21437/INTERSPEECH.2017-35.
- [59] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks, 2014. URL: <https://proceedings.neurips.cc/paper/2014/hash/a14ac55a4f27472c5d894ec1c3c743d2-Abstract.html>.
- [60] Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. Multilingual translation with extensible multilingual pretraining and finetuning, 2020. URL: <https://arxiv.org/abs/2008.00401>, arXiv:2008.00401.
- [61] The Omnilingual MT Team, Pierre Andrews, Mikel Artetxe, Mariano Coria Meglioli, Marta R. Costa-jussà, Joe Chuang, David Dale, Cynthia Gao, Jean Maillard, Alex Mourachko, Christophe Ropers, Safiyyah Saleem, Eduardo Sánchez, Ioannis Tsiamas, Arina Turkatenko, Albert Ventayol-Boada, and Shireen Yates. Bouquet: dataset, benchmark and open initiative for universal quality evaluation in translation, 2025. URL: <https://arxiv.org/abs/2502.04314>, arXiv:2502.04314.
- [62] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971, 2023. URL: <https://doi.org/10.48550/arXiv.2302.13971>, arXiv:2302.13971, doi:10.48550/ARXIV.2302.13971.

- [63] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017. URL: <https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html>.
- [64] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners, 2022. URL: <https://openreview.net/forum?id=gEZrGCozdqR>.
- [65] Wikipedia contributors. Ligurian language — Wikipedia, the free encyclopedia, 2026. [Online; accessed 7-September-2026]. URL: https://en.wikipedia.org/w/index.php?title=Ligurian_language&oldid=1371236201.
- [66] Wikipedia contributors. Republic of Genoa — Wikipedia, the free encyclopedia, 2026. [Online; accessed 7-September-2026]. URL: https://en.wikipedia.org/w/index.php?title=Republic_of_Genoa&oldid=1371581492.
- [67] Wikivivagna. Wikivivagna – a librajna in lengua ligure. Ligurian language edition of Wikisource. URL: https://lij.wikisource.org/wiki/P%C3%A0gina_prin%C3%A7ip%C3%A2.
- [68] Ludwig Wittgenstein. *Philosophical Investigations*. Wiley-Blackwell, New York, NY, USA, 1953.
- [69] Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D. Manning, and Christopher Potts. Reft: Representation finetuning for language models. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL: http://papers.nips.cc/paper_files/paper/2024/hash/75008a0fba53bf13b0bb3b7bff986e0e-Abstract-Conference.html.
- [70] Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. Byt5: Towards a token-free future with pre-trained byte-to-byte models. *Trans. Assoc. Comput. Linguistics*, 10:291–306, 2022. URL: https://doi.org/10.1162/tacl_a_00461, doi:10.1162/TACL_A_00461.

- [71] Yi Yang and Jacob Eisenstein. A log-linear model for unsupervised text normalization. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 61–72. ACL, 2013. URL: <https://doi.org/10.18653/v1/d13-1007>, doi:10.18653/V1/D13-1007.
- [72] Seyma N. Yildirim-Erbasli and Okan Bulut. Exploring data augmentation in a low-resource language context: A case study on text generation for reading comprehension in turkish. *Algorithms*, 19(5), 2026. URL: <https://doi.org/10.3390/a19050413>, doi:10.3390/A19050413.
- [73] Sevinj Yolchuyeva, Géza Németh, and Bálint Gyires-Tóth. Text normalization with convolutional neural networks. *Int. J. Speech Technol.*, 21(3):589–600, 2018. URL: <https://doi.org/10.1007/s10772-018-9521-x>, doi:10.1007/S10772-018-9521-X.
- [74] Vilém Zouhar, Niyati Bafna, Mukund Choudhary, Maike Züfle, Sara Rajaei, Pinzhen Chen, Jannis Vamvas, Sara Papi, Ona de Gibert, Bhavitvya Malik, Eliya Habba, Orfeas Menis Mastromichalakis, Patrícia Schmidtová, Michelle Wastl, Sheriff Issaka, Leshem Choshen, Stella Biderman, Antonis Anastasopoulos, Jan Niehues, Rico Sennrich, Mrinmaya Sachan, Ondřej Bojar, Kenton Murray, Jörg Tiedemann, Alham Fikri Aji, Philipp Koehn, Christof Monz, Alexandra Birch, Sowmya Vajjala, Chalamalasetti Kranti, Cristina España-Bonet, Nobin Sarwar, David Kaczér, Shunta Asano, Malik Marmonier, Daban Q. Jaff, Vaisakhi Mishra, Hend Al-Khalifa, Gabriele Sarti, Sourajit Saha, Nils Rehlinger, Juan Daniel Cuervo Villa, Jonathan Tonglet, Saugata Purkayastha, Dominik Macháček, Jagannathan Ramanujam, Heejin Do, Zuzana Nadova, Fred Philippy, Fabian Retkowski, Maria Lymperaïou, Silvia Casola, Hanna Yukhymenko, Shubhashis Roy Dipta, Sangwon Ryu, Andrés Jerez, Ron Keinan, Shuaib Shuaib Yusuf, Avantica Vempati, Maria Carmen Staiano, Sukannya Purkayastha, Adrian Cosma, Vitalii Babenko, Erivan Inan, Aviral Nigam, Wafa Aissa, Fatima Haouari, Venkata Prasanth Kumar Gummadi, Mehdi Jafarzadeh, Valentin Scourneau, Lukas Edman, Kaiser Sun, Shaomu Tan, Mohammad Sadegh Gholizadeh, Johannes-Rudolf David, Dipankar Srirag, Javier García Gilabert, Ruta Binkyte, Manar Ali, Ana-Maria Bucur, Sabry E. Farrag, Youssef Saber, Yihong Liu, Jean Mailard, Cojocar Nicoleta, Xiaochuang Yuan, Sina Ahmadi, Philipp Mondorf, Kaustubh Dhole, Roman Wixinger, Shenbin Qian, Manuel Tuor, Sergey Troshin, Jonathan Yahav, Fida Mohammad Thoker, Amir Arsalan Rezapour, Lance Calvin Lim Gamboa, Manon Reusens, Kättriin Kukk, Koel Dutta Chowdhury, Giuseppe Gallipoli, Christian Hoang, Shaswati Saha, Seth Aycock, Jan Kocoń, Bo Chen, Linh Vu, Vatsal Venkatkrishna, Arafat Ahsan, Luan Thanh Nguyen, Hassan Soliman, Daryna Dementieva, Theresia Veronika Rampisela, Ngoc Quynh Tram Do, Marius Huber, Kazuki

Egashira, Azmine Toushik Wasi, Vladislav Poritski, Mike Zhang, Deep Shah, Paul Gavrikov, Luis Frentzen Salim, David Africa, R. Damanhuri, Bello Umar Bello, Anumit Garg, Gengyu Rao, Pawan Sasanka Ammanamanchi, Kamile Dementaviciute, Andrianos Michail, L D M S Sai Teja, Dawei Zhu, Yi Fan, Wei Liu, Farhan Farsi, Elias Herranen, Sankalan Pal Chowdhury, Karen Sanchez, Farzad Shami, Ashok Urlana, Zimu Wang, Tomasz Limisiewicz, Priyaranjan Pattnayak, Marii Ojastu, Hongbin Na, Emilian Radoi, Chenyi Zhao, Carlos Hinojosa, Andrea Gregor de Varda, Zaid Alyafeai, Reem Alzahrani, Nehal Kathrotia, Alex Flückiger, Ulysses Sekai Tully Carr, Jimson Paulo Layacan, Guy Kaplan, Ritwik Tiwari, Rishit Dagli, Oksana Volchek, Isaac R Caswell, Bowen Yi, Blanka Kövér, Amir Hossein Yari, Aicha Chorana, Zhengxiang Wang, Selja Keränen, Samuel Simko, Joy Olusanya, Jenny Chim, Enzo Doyen, Vivek Harsha Lakkamaneni, Sophia Conrad, Pouya Sadeghi, Panayiotis Panayiotou, Luis Lara, Jannatul Nayem, Eran Yahav, Debanshu Das, Antonia Karamolegkou, Anmol Goel, Aishik Mandal, Tommaso Cerruti, Raoyuan Zhao, Mykola Haltiuk, Thura Aung, Naser Almousa, Amir Hossein Kargaran, Rachel Bawden, Qiaoyuan Zheng, Mateusz Lango, Beni Egressy, Fidel Rodríguez Velásquez, Natchapon Jongwiriyanurak, Minh Ngoc Do, Marco Gaido, Lena Libon, Dzmitry Kuzmin, Badal Nyalang, Antoine Taroni, Andrei Niculae, Abdulaziz Nura Kani, Rushikesh Zawar, Marek Šuppa, Beatrice Savoldi, Andreas Simons, Rayyan Merchant, Ilai Yaron Levy, Francesco Pinto, Ziyi Yang, Yolanda Xavier, Samuel Frontull, Muhammad Ravi Shulthan Habibi, Kenneth Enevoldsen, Harris Abdul Majid, Francesca Padovani, Tim Graf, Tatiana Bielakova, Sharifa Djurabaeva, Shaoxiong Ji, Raia Abu Ahmad, Pavel Stepachev, Jirui Qi, Ayush Sunil Munot, Alireza Pakniat, Ayla Rigouts Terryn, Yuxing Lu, Yurii Paniv, Xiyang Fu, Tosin Adewumi, Sunisth Kumar, Stéphane J. P. S. Thunus, Shree Harsha Bokkalahalli Satish, Shayan Bali, Prakhar Gupta, Papa Abdou Karim Karou Diallo, Matija Akrap, Marko Culjak, Kristýna Onderková, Joseph Attieh, Esrael Teferi Tensay, Elisabeth Fittschen, Benoît Sagot, Jingwei Ni, and Yu Fan. Last translation benchmark, 2026. URL: <https://arxiv.org/abs/2609.04173>, arXiv: 2609.04173.

- [75] Andrew Zupon, Evan Crew, and Sandy Ritchie. Text normalization for low-resource languages of africa. In Kathleen Siminyu, Julia Kreutzer, Hady Elsahar, Vukosi Marivate, Nishant Subramani, Jade Z. Abbott, and Bernardt Duvenhage, editors, *2nd AfricaNLP Workshop Proceedings, AfricaNLP@EACL 2021, Virtual Event, April 19, 2021*, 2021. URL: <https://arxiv.org/abs/2103.15845>.