UNIVERSITÀ DEGLI STUDI DI PALERMO

Scuola di Dottorato

PhD in Information and Communication Technologies

Dipartimento di Ingegneria

Neural Lemmatization for Early Slavic Texts

PH.D. CANDIDATE

Usman Nawaz

COORDINATOR

Prof. Marco La Cascia

ADVISOR

Assoc. Prof. Liliana Lo Presti

CICLO XXXVIII

ACADEMIC YEAR 2024 – 2025

Acknowledgement

First and foremost, I extend my deepest thanks to my supervisor, Assoc. Prof. Liliana Lo Presti, for her invaluable guidance, patience, and constructive feedback. Her academic advice, careful supervision, and trust in my work have played a central role in shaping this research. I am truly thankful for the time and effort she devoted to helping me develop both this thesis and my confidence as a researcher. I am also sincerely grateful to Prof. Marco La Cascia, the group head, for his leadership and for creating a stimulating research environment. His support throughout my doctoral studies has been highly valuable, and I appreciate the opportunity to work within a group that encouraged academic growth and collaboration.

I am grateful to Marianna Napolitano, Product Owner of WP(4) in the ITSERR project, for her support and countless helpful email exchanges. I appreciate the time and care she dedicated to supporting this work and helping me make progress. I would also like to thank my colleagues, fellow researchers, and lab mates, with whom I shared many discussions, ideas, and challenges during these years. Their suggestions, conversations, and collaborative spirit helped me improve my work and made the research process more engaging.

On a personal note, I owe a special debt of gratitude to my family, whose love, prayers, and belief in me have been the foundation of my strength. I deeply wish that my father could see the colours of this beautiful world again with his own eyes, as he could when I began this journey. His prayers, sacrifices, and presence have remained a constant source of strength for me. I am especially thankful to my younger brother and sister for always standing beside me and giving me courage during difficult moments.

My sincere thanks go to Dr. Aqeel Ur Rehman, whose positivity, encouragement, and valuable advice have always been a source of strength for me. I am also grateful for his endless trip plans, most of which remained pending, except for one memorable trip in 2023. I am also deeply grateful to Sheraz Ahmed for his innovative ideas, cooperation, and generous support.

Finally, I dedicate this work to everyone who believed in me and helped me continue despite the challenges. This thesis is not only the result of academic effort, but also a reflection of the care, patience, and generosity I received from the people around me.

List of Publications

The work presented in this thesis has resulted in the following publications and submissions:

1. Nawaz, U., Lo Presti, L., Napolitano, M., & La Cascia, M. (2024, August). Automatic Lemmatization of Old Church Slavonic Language Using A Novel Dictionary-Based Approach. In International Workshop on Document Analysis Systems (pp. 408-421). Cham: Springer Nature Switzerland.

2. Napolitano, M., & Nawaz, U. (2025). From Characters to Meaning: Challenges and Limitations in the Semantic Analysis of Church Slavonic Texts. *FSCIRE RESEARCH AND PAPERS*, 6, 41–65.

3. Nawaz, U., Lo Presti, L., & La Cascia, M. OldSlavicLemma: A Neural Sequence-to-Sequence Model for Lemmatization of Early Slavic Languages. Accepted with minor revisions for publication in *Natural Language Processing*, Special Issue on Language Models for Low-Resource Languages.

4. Nawaz, U., Napolitano, M., Karafillidis, I., Lo Presti, L., & La Cascia, M. Towards Benchmarking Old Church Slavonic Lemmatization. Accepted for publication in BriGap-3: The Workshop on Bridges and Gaps between Formal and Computational Linguistics, 2026.

Abstract

In this thesis, the task of lemmatization for Early Slavic languages, such as Old Church Slavonic (OCS) and Old East Slavic (OES), is addressed as a problem of end-to-end robustness, conditioned on limited supervision, rich morphology, and significant variability in orthography and editorial practices. Unlike the standard languages used today, Early Slavic texts are typically limited, heterogeneous, and affected by representational errors in digitization, such as inconsistent use of diacritics and font-encoded text, including Unicode Private Use Area (PUA) code points. These factors undermine token consistency, inflate the size of the lexicon, and increase Out-of-Vocabulary (OOV) rates, which negatively affect both wordlist-based and neural character-based approaches and can be difficult to diagnose. The primary objective is to develop a reproducible text-processing pipeline that ensures text consistency and improves lemmatization accuracy, especially under conditions typical of Early Slavic texts, with particular emphasis on the ability to generalize to novel and orthographically diverse texts.

The thesis introduces a newly annotated OCS dataset, which serves as a challenging benchmark for adaptation and robustness testing beyond existing standard resources. The results on this dataset also provide empirical support for the corpus-construction, normalization, and annotation strategies developed in the thesis, showing that newly curated and carefully standardized material is essential for robust lemmatization of orthographically diverse Early Slavic texts.

The work also introduces an encoding-aware standardization layer that detects non-standard Unicode usage and, in a deterministic way, normalizes PUA-dependent characters to stable Unicode equivalents. Standardization is treated as part of the modeling backbone, as it reduces out-of-vocabulary word counts, improves the robustness of frequency statistics used in lexical baselines, and allows cross-comparisons across mixed data. On top of this representation, the study introduces a clear, dictionary-based baseline that automatically infers a type-to-lemma mapping from the training data and treats out-of-vocabulary words in a conservative way. This baseline serves as a reference and diagnostic tool to identify failures caused by limited coverage and ambiguity, while the main comparative evaluation also includes established Universal Dependencies (UD) systems such as Stanza and UDPipe 2.

The key contribution of this study is the introduction of OldSlavicLemma, a neural lemmatizer for low-resource historical languages that does not rely on a dictionary. Instead, the model treats lemmatization as a character-level sequence transduction problem, condi-

tioned on a fixed local context window surrounding the word to be lemmatized. The model consists of a stacked Bidirectional Long Short-Term Memory (BiLSTM) encoder and a Long Short-Term Memory (LSTM) decoder with attention. For lemmatization, the model leverages Cross-Layer Multi-Head Attention (CLMA) to integrate low-level orthography and higher-level abstraction in context. Furthermore, it leverages Encoder-Decoder Cross-Attention (EDCA) to generate lemmas that are context-aware and to resolve ambiguity.

Evaluation is conducted with gold tokenization and the official splits of the corresponding UD treebanks, in order to isolate lemmatization from tokenization and sentence segmentation. In addition, a supplementary raw-text-to-CoNLL-U evaluation is conducted on the Early Slavic UD v2.12 and v2.15 treebanks, where the Stanza tokenizer is used and OldSlavicLemma is applied to its tokenized output. Four experimental settings are considered: (i) in-family evaluation on Early Slavic UD treebanks across different UD versions, (ii) evaluation across language families on the SIGTYP 2024 dataset in the constrained regime, (iii) multilingual evaluation on a curated subset of UD treebanks to assess transfer beyond Slavic languages, and (iv) evaluation on the newly annotated OCS benchmark to assess robustness, adaptation, and generalization beyond existing standard resources.

In the in-family evaluation of Early Slavic languages, OldSlavicLemma outperforms strong baseline systems, including UDPipe 2 and the Stanza lemmatizer in its hybrid configuration, with the largest improvements observed in out-of-vocabulary-rich and orthographically complex settings. In addition, the results show that the improvements arise mainly from better handling of out-of-vocabulary words rather than from simple memorization of frequent word–lemma mappings. Ablation experiments also show that encoder-decoder attention and optimization are crucial for robust performance in low-resource settings.

In conclusion, the experiments support the claim that reliable historical lemmatization requires simultaneous control of representation stability and character-level contextualization, with OOV handling and ambiguity as the primary historical failure modes that should be factored into the evaluation.

Contents

List of Figures

List of Tables

List of Acronyms

BERT Bidirectional Encoder Representations from Transformers.

BiLSTM Bidirectional Long Short-Term Memory.

CLMA Cross-Layer Multi-Head Attention.

EDCA Encoder-Decoder Cross-Attention.

LLM Large Language Model.

LSTM Long Short-Term Memory.

NLP Natural Language Processing.

OCS Old Church Slavonic.

OES Old East Slavic.

OOV Out-of-Vocabulary.

PEFT Parameter-Efficient Fine-Tuning.

PLM Pre-trained Language Model.

PUA Private Use Area.

RNC Russian National Corpus.

Seq2Seq Sequence-to-Sequence.

TEI Text Encoding Initiative.

UD Universal Dependencies.

XML Extensible Markup Language.

Chapter 1

Introduction

This chapter outlines the motivation, challenges, and objectives of the thesis, focusing on lemmatization for Early Slavic languages in historical NLP settings. The chapter begins by motivating the study of Early Slavic languages and their relevance in historical NLP. Then, it introduces the lemmatization problem and briefly reviews the main approaches applied to Early Slavic texts. Finally, it presents the objectives of the thesis and its main contributions.

1.1 Motivation for Studying Early Slavic Languages

Early Slavic languages such as OCS and OES have a rich linguistic and cultural role in the history of Slavic communities. It served as the earliest writing medium for Slavic and became the foundation of religion and cultural tradition in the 9th century with its historical significance [60]. As the earliest attested Slavic literary language, it played a major role in the Christianization of Eastern Europe, a legacy associated with the missionary work of Saints Cyril and Methodius [42]. It preserves important evidence for Indo-European morphology and textual transmission, attracting both historians and computational researchers, which is now accessible via major digitization and treebanking efforts [40, 32]. Beyond their philological value, these corpora are increasingly relevant to computational work on diachrony, typology, and manuscript variation, supporting a growing line of research in historical Natural Language Processing (NLP) [28, 81, 30].

By combining diverse collections and enhancing them with temporal and regional metadata, recent resource-building initiatives have increased the accessibility of Church Slavic materials. To facilitate comparative analysis across Church Slavonic variants and periods, DIACU, for instance, combines several sources into a geographically and diachronically annotated dataset [18]. These initiatives demonstrate both the advancement of digitization and a significant practical constraint: robust computational processing at scale is still challenging, even in cases where texts are available.

The major challenges lie in linguistic, representational, and resource aspects. Church Slavic languages have a rich morphology with syncretisms, stem changes, and irregularities. In addition, orthographic variation is considerable, with a single lemma represented in vari-

ous surface forms, spelling variation being ubiquitous, and the use of diacritics and editorial practices varying. As a consequence of the lack of orthographic standardization in historical languages like Church Slavic, the availability of annotated training data is limited, and textual variation remains high. This makes it hard to apply techniques developed for modern standardized languages and provides a strong case for models with strong sequence inductive bias and attention mechanisms.

Another issue that arises is the representation of text. There are many early Slavic sources that are encoded heterogeneously, meaning that the encoding of the texts can vary. This can include the use of different fonts and the use of Unicode Private Use Area (PUA) code points. Although these encoding practices can ensure that certain distinctions are made, they can create issues when interoperability is needed. This can lead to the splitting of identical symbols when their encodings differ, artificially expanding the vocabulary. DIACU [18] highlights problems related to the use of Unicode PUA code points in ensuring interoperability across digital sources. This, therefore, creates the need for the normalization of the texts [67]. In this thesis, the representation and normalization of the texts are considered to be an integral part of the historical NLP process.

Within this context, lemmatization is a fundamental task that maps a word form to its base form, or lemma. It helps to resolve the problem of sparsity, enables more reliable matching and generalization across inflectional variants, links the lexicon, and allows for search and concordance with a degree of accuracy. Lemmatization is widely used as a normalization step that improves the downstream NLP tasks such as translation [35, 52, 85], information retrieval [49, 11, 105], historical text preprocessing [76, 61], and sentiment analysis [9, 1].

1.2 Lemmatization in Early Slavic Languages

Previous lemmatization systems were largely based on rule-based approaches that relied on the availability of large lexica and expert knowledge about derivational and inflectional patterns [82]. While these systems perform well in resource-rich environments, their performance typically suffers in low-resource settings. Furthermore, such approaches are expensive to develop when resources or expertise are limited. Approaches to lemmatization have also focused on the contextual nature of the task, using feature-rich statistical models to identify appropriate edits [22]. Recent approaches have explored the joint modeling of lemmas and morpho-syntactic tags to leverage their mutual constraints and improve overall accuracy [23, 66]. These approaches suggest that contextual and morpho-syntactic information plays a crucial role in lemmatization. However, they still suffer from sparsity and domain mismatch in historical texts.

With the advent of neural Sequence-to-Sequence (Seq2Seq) models [91], lemmatization was increasingly viewed as a character-level transduction problem, consistent with trends in neural machine translation and sequence modeling [21, 10]. Seq2Seq models have been successful in a variety of NLP tasks, such as abstractive summarization [84], syntactic pars-

ing [25], and morphological reinflection [26, 25]. In lemmatization, authors of [83] were among the early studies to use Seq2Seq models, and Lematus [13] demonstrated that context-aware character-level models outperform rule-based and feature-based approaches. Yet, low-resource and historical settings remain challenging due to orthographic variation and data sparsity.

An influential hybrid approach commonly used in UD pipelines is the Stanford system from the CoNLL 2018 shared task [78], later released as the Stanza toolkit [79], which builds a dictionary based on a character-level Seq2Seq model in backoff mode, in addition to a lexicon lookup. This approach is successful across a wide range of UD treebanks but is limited by the availability of a good lexicon and performs poorly in the presence of frequent misspellings. Prior work specific to OCS includes a hybrid lemmatizer [3], while string-similarity measures have also been explored for evaluating OCS lemmatization [4]. On the opposite end of the spectrum, dictionary-based approaches directly induce a word-lemma lexicon from annotated data and rely on lookup without grammatical rules [68]. Such approaches perform well in cases of good lexicon availability but cannot exploit contextual information to disambiguate words and perform poorly on unseen spellings and inflectional forms.

Another major challenge is evaluation. NLP has been greatly facilitated by the availability of standardized benchmarks such as GLUE [100], SuperGLUE [99], and XGLUE [57] for modern languages. However, the situation is considerably more complex for ancient languages. For OCS (`chu`) and OES (`orv`), complex morphology and writing systems require specific evaluation conditions. At the same time, the available material is limited to a few UD treebanks [27] and digital editions, which limits the use of large pretrained models. In a cross-variant retrieval task between *chu* and *orv* languages, [56] showed that classical string matching can outperform word embeddings on short texts. Another problem concerns the behavior of tokenizers when applied to ancient languages. The retrieval task, however, is not the main focus of the present thesis.

1.3 Thesis Objectives

This thesis treats Early Slavic lemmatization as an end-to-end robustness problem. The main idea is that achieving robust modeling requires (i) an encoding-aware standardization component to minimize representational heterogeneity and (ii) lemmatization models that generalize well under domain shift.

A further motivation for this thesis is the lack of newly curated, evaluation-oriented lemmatization resources for OCS beyond the standard benchmark treebanks. While existing resources are highly valuable, they do not fully capture the orthographic diversity, encoding instability, and source heterogeneity found in newly collected material from portals and scholarly editions. For this reason, the thesis also aims to study how additional annotated OCS material can support evaluation of robustness, adaptation, and out-of-vocabulary generalization under realistic historical conditions.

The thesis pursues the following objectives:

1. To design and evaluate an encoding-aware standardization procedure for Early Slavic texts that detects PUA code points, maps them to stable Unicode forms where appropriate, and ensures consistent token identity.

2. To implement a reproducible lemmatization pipeline with transparent lexical baselines and strong comparative systems, enabling systematic comparison between dictionary-based memorization, hybrid UD lemmatization systems, and dictionary-free lemmatization.

3. To develop and evaluate OldSlavicLemma, a dictionary-free character-level encoder–decoder lemmatizer with local contextual conditioning, stacked BiLSTM encoding, CLMA, and EDCA.

4. To assess whether character-level contextual lemmatization improves generalization to OOV forms, orthographically variable forms, ambiguous surface forms, and cross-domain Early Slavic material compared with dictionary-based baselines, Stanza, and UDPipe 2.

5. To construct, validate, and evaluate a newly annotated OCS lemmatization benchmark in order to evaluate robustness, in-domain adaptation, and generalization beyond existing standard resources.

Although lemmatization supports downstream tasks such as search, concordance, and corpus exploration, the main focus of this thesis is token-level lemma prediction and its robustness under historical variation.

1.4 Thesis Contributions

This thesis addresses the challenges of the robust lemmatization methods for Early Slavic texts under conditions of sparse supervision, orthographic variation, and encoding instability. Existing dictionary-based and hybrid UD systems perform well when lexical coverage is high, but they are less effective when the test data contain many unseen forms or representation-level inconsistencies. The thesis contributes methods and resources designed to make these failure modes explicit and to reduce their effect.

The main contributions are as follows:

1. A representation-aware standardization workflow for Early Slavic texts. The workflow detects PUA usage and other non-standard character practices and maps them to stable Unicode representations. This reduces artificial vocabulary fragmentation, improves token consistency, and supports fair comparison across dictionary-based and neural lemmatization systems.

2. A reproducible lemmatization pipeline with transparent lexical baselines. The dictionary-based baseline provides an interpretable reference point for measuring the effects of lexical coverage, ambiguity, and out-of-vocabulary forms. This makes it possible to distinguish memorization-based performance from genuine generalization.

3. OldSlavicLemma, a dictionary-free character-level encoder–decoder lemmatizer for Early Slavic languages. The model uses local contextual windows, stacked bidirectional LSTMs, cross-layer multi-head attention, and encoder–decoder cross-attention to generate lemmas from character-level and contextual evidence.

4. A broad empirical evaluation against strong comparative systems, including Stanza and UDPipe 2, on Early Slavic UD treebanks, SIGTYP 2024 historical data, and multilingual UD data. The evaluation includes overall performance, OOV analysis, ambiguity analysis, and ablation experiments.

5. It introduces an annotated OCS benchmark dataset for lemmatization and demonstrates its value as a challenging evaluation resource beyond the existing OCS resources.

1.5 Overview of the Thesis

The remainder of this thesis is organized as follows.

Chapter 2 presents the historical and methodological foundations of the study. It reviews the main digital resources available for Early Slavic languages, discusses the encoding-related challenges that arise in historical text processing, and situates the task within the broader context of historical and low-resource NLP, with particular attention to lemmatization and domain adaptation.

Chapter 3 describes the data pipeline developed in this thesis. It covers corpus acquisition, unified representation, cleaning, tokenization, and encoding standardization, showing how heterogeneous digital sources are harmonized and converted into interoperable formats suitable for downstream NLP.

Chapter 4 introduces the gold resources developed for Early Slavic lemmatization. It defines the annotation scope, explains the representation formats used for interoperability, presents the annotation guidelines and semi-automatic workflow, and reports dataset statistics together with validation procedures.

Chapter 5 presents the proposed neural lemmatizer, OldSlavicLemma. It formulates lemmatization as a character-level sequence transduction task and describes the model architecture, training objective, optimization strategy, and reproducibility considerations.

Chapter 6 reviews the comparative systems and baselines used throughout the thesis. These include the dictionary-based baseline, UDPipe 2, Stanza, and the shared-task systems considered in the SIGTYP 2024 comparison.

Chapter 7 defines the experimental protocol. It describes the datasets and splits, specifies the evaluation metrics, and introduces the robustness-oriented analyses used to assess

performance under conditions of lexical sparsity, ambiguity, and domain shift.

Chapter 8 reports the experimental results and provides their analysis. It presents findings on Early Slavic treebanks, multilingual and historical benchmarks, newly annotated OCS material, ablation studies, and error categories such as out-of-vocabulary and ambiguous forms.

Chapter 9 concludes the thesis by summarizing the main contributions, discussing the principal limitations of the study, and outlining directions for future research.

Chapter 2

Historical NLP and Foundations

This chapter provides the historical and methodological background for the thesis. It first discusses the main digital resources for Early Slavic languages, including treebanks, editions, and digital collections, and points out the interoperability problems that arise from heterogeneous sources. The chapter discusses problems related to the encoding of historical Slavic languages, in particular, the use of PUA characters and script variation.

Early Slavic lemmatization is positioned in the wider framework of NLP involving historical and resource-constrained scenarios, where data scarcity, orthographic diversity, and tool deficiencies have significant implications on the accuracy. The chapter provides an overview of the most important approaches to lemmatization, including rule-based, statistical, hybrid, and Seq2Seq neural models. The importance of character-based modeling is highlighted in the case of highly inflected languages, which have undergone considerable orthographic changes. In addition, the chapter covers Pre-trained Language Models (PLMs), attention, and adaptation mechanisms for historical text processing. Robust outcomes can be achieved by taking into account all three factors simultaneously—representation, normalization, and modeling.

2.1 Early Slavic in Digital Form

The textual traditions of early Slavic languages, including OCS and OES, have been transmitted in manuscript form over the course of centuries. Today, these texts are made accessible in digital form. From the perspective of NLP, it is useful to identify two categories of resources: (i) digitized resources and scholarly editions that preserve the original layout and formatting used in the publication process, and (ii) linguistic resources that have been annotated in a format useful for training and testing models.

An important step in the history of computational processing of early Indo-European Bible translations is the PROIEL project, which supplies aligned and syntactically annotated texts and serves as a standard resource for OCS in the UD project [40]. Simultaneously, resources for OES were created in the form of the Tromsø Old Russian and OCS Treebank (TOROT), which extends the coverage to include diachronic texts and facilitates comparison

across genres and publication traditions [32]. These resources have made early Slavic languages part of multilingual benchmarks and shared-task settings. At the same time, these resources also highlight the constraints that were imposed on historical NLP, including limited supervisory signals in comparison to contemporary languages, widespread orthographic variation, and the importance of digitization and publication processes in determining the learnability of a model.

Resource development has also expanded beyond the simple single-treebank configuration to incorporate aggregation over variants, centuries, and sources. An exemplary case of this aggregation process is the DIACU resource, which aggregates multiple corpora of Church Slavonic texts into a diachronically and geographically annotated resouce, aiming at supporting comparative studies over time periods and regional redactions [18]. Although such aggregation increases the scope of the resources, making them more useful for a wider range of linguistic studies, it also increases the problem of interoperability, as the diversity of the source materials increases the diversity of encoding, typography, and editorial practices. Thus, current historical NLP approaches treat representation choices as part of the scientific object, rather than a secondary concern.

Evaluations conducted on ancient and historical languages' systems increasingly exhibit a systematic approach. The need for standardized configurations and similar protocols for ancient language corpora is on the rise, accelerating the creation of systems specifically designed for the task. This trend is reflected in the demand for the SIGTYP 2024 shared task [28]. Evaluation results show that performance is influenced not only by linguistic generalization but also by representation and normalization processes, which vary substantially.

2.2 Encoding Challenges

Unlike modern web typography, early Slavic digital resources often preserve their typographic and editorial differences, which may not always be supported by standard Unicode. As a result, the resources may use a mixture of standard Cyrillic characters, combining diacritics with inconsistent composition, mixed-script substitution, and characters from the PUA, whose interpretation may depend on the fonts used or local conventions [8, 50, 15]. In practice, two characters may look identical but, in reality, have different code points, thereby violating a basic NLP assumption that "the same symbol" corresponds to "the same token."

This heterogeneity does not simply manifest as an engineering problem but also directly affects NLP pipelines. Tokenization, dictionary lookups, and even character transduction are sensitive to code point identity. When a character has a representation both in standard Unicode and in the PUA, frequency statistics are split, vocabulary size increases artificially, and OOV rates increase unnecessarily even though the linguistic content has not actually changed. These effects are particularly problematic for low-resource languages, where every additional variant of representation makes the problem worse by reducing the amount of available supervision data. In the corpus aggregation effort for Church Slavonic, DIACU highlights the

use of PUA and the issue of visualization as a problem for practical use, emphasizing the need for a systematic approach to character mapping and normalization as a necessary step for reusing data between portals and editions [18]. Similar issues are raised in the discussion of Church Slavonic typography and standardization, including the use of fonts, ad hoc historical encoding schemes, and the mismatch between rendering and Unicode code point identity [8, 50].

This interaction between representation and modeling is particularly clear in the process of lemmatization, where dictionary-based methods might fail silently if the variation occurs only at the level of the encoding, and where neural models of characters might learn incorrect relationships based on the use of different font sets rather than reflecting true morphological relationships. Evaluation can also be misled if the test data and the system output have different normalization or code point strategies, leading to a preference for encoding compatibility over actual linguistic ability. These issues underpin the use of encoding-aware methods as a necessary step for any kind of reproducible historical NLP and provide the rationale for the standardization layer introduced later in this thesis [8, 15].

2.3 NLP for Historical and Low-Resource Languages

Historical NLP sits at the intersection of low-resource learning and domain adaptation. In comparison to contemporary standardized languages, historical corpora are small, noisy, and heterogeneous. They have long-tailed distributions of rare words, sparse annotation, and great variation due to temporal, geographic, genre, and editorial factors. This increases sparsity and makes generalization over unseen words the main source of errors in a variety of tasks.

One practical consequence is that successful methodologies in high-resource environments often require adaptation or the addition of inductive biases to ensure stability. The SIGTYP 2024 benchmark set is a good example that reflects the opportunities and limitations of using contemporary architectures in historical data environments [28], while pretrained multilingual transformers have a chance to perform competitively in unconstrained environments, their relative strength may diminish in resource-constrained environments, and issues with instability may occur. Research discussed in SIGTYP and related studies emphasize that successful historical NLP pipelines often combine multiple components, including careful preprocessing, parameter-efficient adaptation, and architectures that balance representation and inductive biases [30, 81, 41].

For Early Slavic languages, the aforementioned pressures are compounded by the instability of orthography and the diversity of encoding. Therefore, a robust processing strategy should treat representation, normalization, and modeling as a cohesive problem, rather than as individual stages of a pipeline. This thesis follows this approach, viewing encoding standardization as supporting infrastructure, lemmatization as the primary modeling objective, and evaluating the proposed approaches under simulated conditions of domain shift and high OOV frequency.

2.3.1 Low-Resource NLP as a Context for Historical Lemmatization

The Early Slavic lemmatization task addressed in this thesis can be situated within the broader methodological literature on low-resource NLP. Low-resource settings are typically characterized by limited annotated data, weak tool support, and high costs for preprocessing, normalization, and annotation [73]. These conditions are directly relevant to historical language processing. Although Early Slavic texts raise additional philological and editorial issues, they share core practical constraints with other low-resource NLP scenarios, especially data scarcity, orthographic variation, limited interoperability, and the need for careful resource preparation.

A recurring observation in low-resource NLP is that model improvements alone cannot compensate for the absence of reliable linguistic resources and interoperable preprocessing pipelines [73]. This has been shown across several tasks and languages, including part-of-speech tagging for Khasi and Mizo [102, 70, 65], dependency parsing for Xibe [106], question answering for Swahili [101], hate speech detection for Brazilian Portuguese [95, 96], and machine translation for low-resource languages [45, 47]. These studies show that the availability of texts alone is not sufficient: the texts must also be computationally usable, consistently encoded, and supported by appropriate annotation or preprocessing tools.

Annotation cost is another point of contact between low-resource NLP and historical NLP. Developing task-specific resources for low-resource languages often requires close collaboration with domain experts and careful validation [73]. This is visible in corpus and annotation work for Khasi POS tagging [102, 65], Mizo POS tagging [70], Swahili question answering [101], and hate speech resources for Brazilian Portuguese [95]. In historical language processing, this requirement is even stronger because annotation decisions may depend not only on grammar and vocabulary, but also on editorial conventions, spelling variation, manuscript traditions, and source-specific practices. The resource development strategy adopted in this thesis therefore combines existing resources, harmonization, semi-automatic annotation, and expert correction, as described in Chapter 4.

At the modeling level, low-resource NLP includes a range of approaches, from rule-based and lexical methods to statistical, neural, and transformer-based systems [73]. Existing work includes CRF-based models [70], BiLSTM and CRF-BiLSTM systems [102, 12], transformer-based transfer methods [65, 74], and task-specific neural approaches for machine translation [47, 45]. For historical lemmatization, lexical and rule-based systems remain useful because they are interpretable and make coverage errors explicit. However, character-level neural methods are especially relevant when the goal is to generalize to unseen forms, spelling variants, and morphologically complex word forms. This thesis follows this rationale by using a dictionary-based baseline as an interpretable reference point and a character-level neural model for dictionary-free generalization.

Historical lemmatization also differs from many contemporary low-resource NLP tasks because scarcity is intertwined with textual transmission. In modern low-resource settings, the main problem is often the limited amount of annotated data. In historical settings, scarcity

is also shaped by diachronic variation, redactional diversity, scribal variation, edition-specific conventions, and unstable encoding practices. This makes Early Slavic lemmatization not only a low-resource learning problem, but also a problem of representation and philological consistency.

However, it is also beneficial to distinguish lemmatization from stemming, especially because the latter is often considered a member of the former's broader family under the label of "text normalization." Stemming generally uses rules based on the removal of affixes for the normalization of texts and does not necessarily ensure the resulting word is a member of the lexicon. Lemmatization, on the other hand, aims at retrieving a dictionary form (lemma) using a variety of tools and information sources, especially lexical ones and often accompanied by the use of other morpho-syntactic information such as POS tags [103, 46]. In texts with a complex morphology or a complex history, the consequences of these processes can be significant, the former may cause the conflation of forms with different meanings or uses, while the latter may not perform well with a variety of spellings or a limited presence in the lexicon.

2.4 Lemmatization: Rule-Based, Statistical, and Neural Approaches

The process of lemmatization has been a research interest in NLP for the last few decades [39]. Most computational models have traditionally relied on rule-based approaches that used morphological grammars and lexicons to perform lemmatization [33, 68]. While these models have demonstrated high accuracy in controlled environments with adequate resources and standardization, their effectiveness is often dependent on the availability of resources and tends to degrade in the presence of orthographic variation, frequent irregularities in inflectional paradigms, or inadequate documentation—common characteristics associated with historical texts such as OCS.

Statistical and machine learning techniques have also offered more flexible approaches with the introduction of data-driven transformations instead of grammars. Finite state transducers and probabilistic systems aided the learning of lemmatization patterns from the data rather than rules [31, 38]. However, a large number of these systems were still language-specific and suffered from sparsity.

A popular technique in the context of machine learning-based lemmatization is edit tree classification, in which word-to-lemma transformations are identified as edit rules, and a classifier is used to choose the appropriate rule at test time [88, 19]. These models generalize to new words as long as the relevant edit pattern is represented in the training data. These models can also leverage contextual information to disambiguate, such as through the use of global sentence information [66] or contextualized representations. In settings with substantial orthographic variation and limited training data, the edit rule representations can be limited, and OOV error rates are the dominant performance factor.

Neural architectures have redefined lemmatization as a character-level sequence transduction. With the advent of Seq2Seq models [91], lemmatization has been able to fit into a larger class of transduction models. This class of models has also included neural machine translation [21, 10], summarization [84], syntactic generation/parsing [25], and morphological reinflection [26, 25]. Initial models have been able to demonstrate the effectiveness of applying the seq2seq model to monotone string transformation settings [83]. This has been particularly relevant to lemmatization. Furthermore, the addition of contextual information has been able to provide substantial improvements, especially when morphology is complex and ambiguity is common [13].

A very influential model has been the hybrid model proposed for the CoNLL-2018 task [78] and has been released as the open-source library Stanza [79]. This model has been able to combine a dictionary-based model and a neural character-level seq2seq model. This has been able to provide a robust and widely adopted baseline. On the other hand, a purely dictionary-based model has been able to provide competitive results for out-of-context spellings [68]. For example, a model has been able to induce a word-lemma lexicon on annotated data and perform lookups. This model has been able to operate without the use of grammatical rules. This model has been able to provide a very interpretable and useful model for philological research. However, the model has been limited in terms of coverage and has not been able to make use of contextual information to disambiguate homographs and out-of-lexicon words.

Finally, Seq2Seq modeling in neural networks is related to morphological reinflection, which can be considered the inverse of lemmatization. In fact, tasks such as CoNLL- SIGMORPHON 2017 have fostered models that predict form-to-form relationships given morphosyntactic features [25]. Several competitive systems have employed neural transducers to carry out the task [98, 14], and even joint or multitask training with related tasks, such as tagging, lemmatization, and reinflection, has been explored to boost robustness [71].

2.5 Pretrained Language Models and Domain Adaptation for Historical Text

The development of PLMs has changed the broader NLP landscape by shifting from sparse lexical representations toward dense and contextual representations. Standardized benchmarks such as GLUE [100], SuperGLUE [99], and XGLUE [57] have helped establish common evaluation protocols for modern languages. For ancient and historical languages, however, comparable benchmarks remain more limited, and recent shared tasks such as SIGTYP 2024 have emphasized the importance of evaluation protocols that account for limited data, heterogeneous resources, and preprocessing effects [27, 28].

The traditional pipeline in modern NLP involves sparse representations such as one-hot or bag-of-words representations coupled with linear or probabilistic sequence models. One major shift in the field was the introduction of distributional representations in the form of word embeddings. These representations map words to dense vector representations based

on large-scale corpora. Word2Vec and GloVe have been prominent in this approach, allowing neural models to process words in a meaningful manner rather than relying on sparse representations [34, 75]. However, these representations have limitations in that each word is associated with a fixed vector regardless of the context in which it is used. Furthermore, OOV words are not represented in any meaningful manner.

2.5.1 From sparse lexical representations to Contextual embeddings

Neural models of NLP require numerical representations for input data. One basic approach to numerical representations is one-hot encoding for this method, each word in the vocabulary is represented using a binary vector of length $|V|$ with one active position corresponding to that word type. One-hot encoding is simple, and deterministic representations are also very sparse and cannot capture any similarity in meaning between words. Semantically similar words are not closer in one-hot space than dissimilar words. One-hot representations also lack any means to capture any distributional properties or meaning.

A major breakthrough in NLP came with the introduction of "distributional word embeddings," in which each word type is associated with a dense vector representation in a lower-dimensional space that is learned from the statistics present in the corpus. Unlike one-hot encoding, in which words are represented using symbolic representations that are distinct from one another, embedding representations place words that have similar usage patterns, tend to place close to one another in vector space, capturing important syntactic and semantic patterns [34, 75]. This marked a shift in the foundation on which current nlp pipelines were built before the widespread adoption of PLMs.

One of the most influential static word embedding techniques is word2vec. This method uses shallow neural networks to learn word representations based on local context [62, 63]. The two main architectures of word2vec are Continuous Bag of Words (CBOW) and Skip-gram. In the former, the model predicts a word based on the surrounding context, whereas in the latter, it predicts the surrounding context words based on a target word. As a result of the prediction-based optimization, word2vec often captures relational information in the word vector space.

Another approach to word representation is the use of the complementary method known as GloVe [75], which uses global word-word co-occurrence statistics to create word vectors. While word2vec can be viewed as a model for local context prediction, the GloVe approach can be viewed as a direct factorization of the information contained in the co-occurrence matrix, with the goal of capturing local as well as global distributional information. In fact, both sets of approaches have been shown to produce effective word representations for use in a variety of downstream NLP tasks.

FastText is highly relevant in morphologically rich and low-resource settings, as it represents words based on character n-grams, rather than using each word as a unit [16]. This is highly beneficial in that it improves robustness to rare words and OOV words, as the representation of a word can be formed using the observed subword units, even if the full word

was never observed during the training process. This is particularly useful in settings that exhibit a lot of variation in words, such as in historical texts, where many words may appear infrequently but have similar stems or suffixes.

Despite these advantages, word2vec, GloVe, and fastText, in their standard settings, are *static* word representation models, where a word type is represented by a dedicated vector irrespective of the sentential context in which the word appears. This is often a highly limiting assumption, as words are often context-dependent and polysemous. For example, the word *bank*, used in a financial context and a riverine context, is represented by the same lexical vector in static word representation models. This is a major limitation of static word representation models, which is of particular interest to historical NLP and lemmatization.

Static embeddings face the OOV problem, meaning that if a word was not seen during the training process, no dedicated vector representation will be learned for that word. Sub-word embeddings [16] attempt to mitigate the OOV problem. A notable method that achieves this is fastText, which uses n-gram representations of the word's characters. This approach ensures that any word, even if not seen during the training process, can still benefit from a vector representation. Meanwhile, the development of sub-word level tokenization techniques such as Byte-Pair Encoding (BPE), WordPiece, and SentencePiece has enabled the implementation of open-vocabulary modeling for historical languages, given the fact that many infrequent or OOV surface forms exist for many familiar lemmas [85, 53, 86].

Contextual embedding models have been developed to tackle the major drawback of static embedding models. These models have the ability to produce contextual representations that are conditioned on the surrounding sentence context. However, subword modeling also fails to tackle this problem on its own. Transformer models have been identified as a powerful architecture to contextualize representations using self-attention. This enables any position in the sequence to attend to any other position in the sequence. This property enables the modeling of any dependency without using recurrent architecture [97]. For practical implementation, Transformer contextualization involves stacked layers that combine multi-head self-attention and position-wise feed-forward layers along with positional encoding to capture sequence order [97]. This property of contextualization coupled with the high computational parallelism has made Transformer the state-of-the-art architecture for PLMs.

2.5.2 Attention Mechanisms and Transformer Architecture

Attention mechanisms were first introduced in neural Seq2Seq models to allow the decoder to dynamically access relevant parts of the encoded source sequence instead of relying on a single fixed-size representation [10, 59]. This idea proved especially effective in machine translation and later influenced a broad range of NLP tasks. Transformer models extend this principle by replacing recurrence with architectures built around stacked self-attention and feed-forward layers, allowing positions to model dependencies across the sequence directly [97]. In self-attention, each token is mapped to query, key, and value vectors, and attention weights are computed from query–key similarity, normalized with a softmax, and used to

form a weighted sum of value vectors. Multi-head attention repeats this process in parallel across several learned subspaces, which improves representational flexibility [97].

This concept has had a significant influence outside the realm of machine translation, and it has been applied to many different NLP tasks, such as tagging, parsing, and sequence labeling, due to the improvement that this approach offers to the model's focus on the most critical parts of the input at any particular prediction step [10, 59]. In the context of historical and morphologically rich languages, this particular aspect of the input is significant, as the information that is critical for a particular prediction can, at times, appear at multiple locations.

A central innovation of the Transformer architecture is self-attention. In this case, a sequence attends to itself; unlike encoder-decoder attention, self-attention models relationships among tokens within the same sequence. This generates contextual representations where each token in the input sequence receives information from other tokens [97]. This concept is particularly useful due to its ability to handle long-range relationships without relying on recurrence.

The Transformer architecture, as proposed in [97], replaces recurrent and convolutional sequence-modeling components with layers built around self-attention and position-wise feed-forward networks. In the traditional encoder-decoder architecture of the Transformer, the encoder stacks multi-head self-attention and position-wise feed-forward layers, while the decoder stacks masked self-attention, encoder-decoder attention, and feed-forward layers. This design allows for a high degree of parallelization, which is a major factor behind the Transformer's success in NLP.

At the core of Transformer attention is the mapping from a set of *queries* to corresponding *keys* and *values*. Given a query matrix $\mathbf{Q}$, a key matrix $\mathbf{K}$, and a value matrix $\mathbf{V}$, scaled dot-product attention is defined as:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}, \tag{2.1}$$

where d_k is the dimensionality of the key vectors [97]. The dot products $\mathbf{Q}\mathbf{K}^\top$ measure compatibility between queries and keys, the softmax normalizes these scores into attention weights, and the weighted sum over $\mathbf{V}$ produces the output representations. The scaling factor $1/\sqrt{d_k}$ stabilizes optimization by preventing the dot-product magnitudes from becoming too large.

In practice, Transformers use *multi-head attention*, in which attention is computed in parallel over multiple learned projection subspaces and the resulting head outputs are concatenated and linearly transformed [97]. This allows the model to capture different types of relationships simultaneously (e.g., local agreement, long-distance dependencies, or structural cues) rather than relying on a single attention map.

Because self-attention is permutation-invariant, positional information must be injected explicitly. The original Transformer adds *positional encodings* to input embeddings so that

the model can represent token order [97]. In the sinusoidal formulation:

$$PE(\text{pos}, 2i) = \sin\left(\frac{\text{pos}}{10000^{2i/d_{\text{model}}}}\right), \qquad PE(\text{pos}, 2i+1) = \cos\left(\frac{\text{pos}}{10000^{2i/d_{\text{model}}}}\right), \tag{2.2}$$

where pos denotes the token position, i indexes the embedding dimension, and d_{model} is the model dimension [97]. These encodings provide a deterministic representation of absolute position and also support learning of relative positional relationships.

In the current thesis, the main importance of attention and Transformers lies in the methodological, rather than architectural, role in supplying the underlying principles of modern contextual representations and sequence transduction models. While the final lemmatization model may not necessarily be a Transformer, attention mechanisms will inevitably remain a key player in disambiguation, leveraging local contexts and improving robustness to spelling variations in historical texts.

2.5.3 PLM families and Adaptation strategies

PLMs based on the transformer architecture can be classified according to their architecture and training objectives. For example, encoder-only model uses bidirectional contextual representations, and their training objective is masked language modeling. Bidirectional Encoder Representations from Transformers (BERT) is a landmark model that has proven that such models can be transferred to a wide range of discriminative tasks after fine-tuning [29]. Another model, RoBERTa [58], shows that choices of optimizer and training data size can lead to substantial improvements over the original BERT setup. For the multilingual setup, multilingual BERT [77] and XLM-R [36] have proven that a wide range of languages can be used to train a model, which is beneficial but poses a risk of a mismatch between the variety of the training and test sets, especially if the variety of the test set is significantly different from that of the training set.

Encoder-Decoder Transformer architectures consist of an encoder for contextualizing the input sequence and a decoder for generating the output sequence in an autoregressive manner with cross-attention for conditioning the generation process based on the encoder states [97]. This setup is particularly suitable for Seq2Seq tasks like machine translation and other conditional generation problems [10, 59, 91]. In morphological transduction tasks, including lemmatization defined as character-level sequence generation, encoder-decoder architectures have been found to capture productive transformations rather than simply relying on memorization and lookup tables [83, 13, 48].

Decoder-only Transformers apply causal self-attention so that each position attends only to previous positions, supporting autoregressive next-token prediction. This design underlies Large Language Model (LLM), which are trained to model $p(w_1, \ldots, w_n) = \prod_i p(w_i \mid w_{<i})$ and are widely used for generation and instruction-following after additional alignment steps [80, 17, 72, 2]. For historical varieties, their practical usefulness depends on whether script, orthography, and domain are represented in pretraining, and whether tokenization yields sta-

ble segmentation.

Despite the strong contextual representations that PLMs provide, the historical corpora show systematic differences compared to the pretraining data. They are small, heterogeneous, and the variation caused by factors such as chronology, region, genre, scribal practices, and editorial practices creates significant domain shifts and long-tailed distributions for infrequent phenomena. Therefore, transfer learning can work well, but it may become brittle under sparse supervision [28, 81, 41]. In the case of Early Slavic, the representational heterogeneity of sources makes the situation worse, as inconsistent use of diacritical marks and encoding can introduce false effects of OOV rates, even when the linguistic form remains the same [18].

A commonly used method to deal with domain shift is fine-tuning the pre-trained models on in-domain data. In a low-resource historical setting, fine-tuning might be unreliable and prone to overfitting, which leads to the application of methods that emphasize inductive bias and Parameter-Efficient Fine-Tuning (PEFT) [30, 81, 41]. PEFT reduces the cost of the training process by fine-tuning a limited number of parameters. One of the commonly used methods of PEFT is the application of adapter modules, which consist of the addition of a number of layers to a pre-trained model, which is subsequently fine-tuned on the target data, often improving the results while avoiding the problem of catastrophic forgetting [44]. For example, a study by TartuNLP [30] applied the PEFT method using the adapter modules [44] to the XLM-RoBERTa model [24] in a historical shared task setting. Other methods have focused on the application of character-aware architectures that can better handle the complexities of orthography and morphology, such as hierarchical transformer architectures [81]. Overall, the results of the experiments presented in the SIGTYP 2024 shared task [41, 28] indicate that a hybrid approach, such as the application of a recurrent encoder with an attention mechanism [6, 5], can be competitive in the field of low-resource historical NLP, similar to the results obtained.

With regard to historical text processing, representation mismatches may be the main performance factor. For example, generic tokenizers may split a word into many rare subwords for historical spellings, and heterogeneous representations may cause two strings that look the same at the surface level to be considered different. For Church Slavic variants, the results of cross-variant retrieval experiments suggest that classical string processing methods may perform better than off-the-shelf multilingual embedding models on short texts where the tokenizer and embedding representations are not well aligned with chu/orv orthography [56]. Though not the main topic of the thesis, the above discussion serves to illustrate a general methodological consideration with direct implications for lemmatization, without representation-aware preprocessing and evaluation, the results may not generalize well at the linguistic level but rather at the level of encoding compatibility and tokenizer behavior.

In lemmatization, domain shift and representation mismatch typically appear as an OOV-heavy error profile and an unstable pattern of generalization to unseen spellings. The current UD pipelines provide robust baseline performance through a hybrid approach combining a

dictionary lookup with a neural backoff method [78, 79]. The edit rule approach maintains competitive performance in UD settings [89, 90, 87]. In historical texts, in which significant spelling variation and limited data availability are expected, character-level transduction remains a successful inductive bias, especially in cases where contextual cues aid disambiguation [13, 83]. Therefore, in the current thesis, lemmatization in Early Slavic texts is treated as an end-to-end robustness problem, in which encoding-aware standardization helps to ensure token consistency across disparate sources [18], while model performance is evaluated in a way that incorporates OOV patterns and domain shift, in accordance with current approaches to historical benchmarking [28].

Chapter 3

Data Resources, Corpus construction, and Encoding Standardization

In this chapter, the description of the data-preparation pipeline designed for Early Slavic lemmatization is presented. This pipeline covers both text acquisition from treebanks and scholarly editions and portal collections, as well as the process of normalization of these resources, proving that while source collection is indispensable in order to improve the scope and supervision, it simultaneously results in significant interoperability problems related to diverse encoding systems and different editorial standards. One of the contributions discussed in the chapter is an encoding-aware standardization layer that can detect PUA and other unconventional Unicode usage and convert it into proper Unicode representations. As a result, it helps avoid unnecessary vocabulary expansion, stabilizes token identity, and supports reliable processing by different methods (both dictionary-based and neural). The chapter also describes unified representation, cleaning, tokenization, and conversion into interoperable formats suitable for subsequent processing and evaluation. These steps serve as the basis for the subsequent research chapters.

3.1 Corpus, Resources, and Acquisition

One of the primary practical challenges faced in the development of the Early Slavic NLP system is the lack of a single source that offers sufficient lemma-annotated material with broad genre, redaction, and orthography coverage. This leads to the training data for the lemmatization module being limited in quantity as well as scope, with individual sources or treebanks covering small facets of the total linguistic spectrum. To increase the training data for the lemmatization module, texts are aggregated from multiple online sources such as UD treebanks, scholarly sources, and portal-based sources. This is done for two primary reasons. First, it allows the model to use the greatest quantity of lemma supervision possible. Second, it allows the model to use realistic variation, which is a necessity when assessing robustness in historical contexts.

At the same time, the aggregation of sources is also the primary reason for the significant

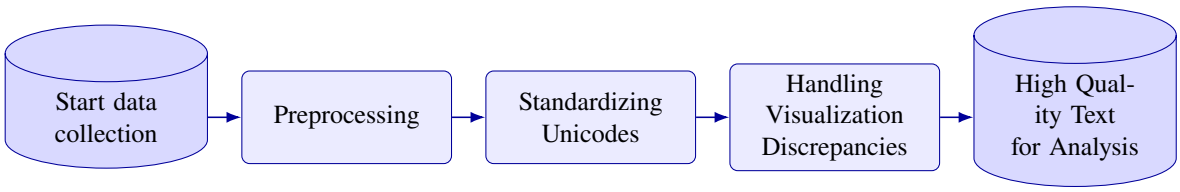

Figure 3.1: End-to-end workflow from acquisition to interoperable text suitable for downstream NLP.

interoperability challenges that are faced in this system. This is because the web and edition pipelines often use non-standard encoding to maintain typographic distinctions, even when two strings are visually indistinguishable. If not incorporated, this will cause issues with the vocabulary, OOV tokens, and dictionary lookups, thus causing problems with the annotation as well as the model. In this regard, the aggregation of the sources is complemented by the standardization of the encoding, as well as the harmonization of the corpus, as discussed (Sections 3.2–3.3).

Figure 3.1 shows the entire process of the corpus pipeline, including acquisition, harmonization, cleaning, standardization of encoding, and conversion into an interoperable format for further processing with various NLP techniques.

The training and testing of the lemmatization model relies on the annotation of the lemma at the token level, which in turn limits the available resources for the task. Hence, the primary source of OCS texts provided by PROIEL and the UD version of the same is exploited in the experiments, as it provides tokenization, sentence segmentation, and linguistic annotation that can be reproduced in the modeling process [40]. To conduct the robustness experiments in a controlled setup, OES texts are also utilized when the tokenization and annotation conform to the UD guidelines, allowing the investigation of domain adaptation from one historical variety to another [32].

To utilize other types of resources that do not conform to the treebank style, portal-based aggregated datasets that offer variation in time, space, redaction, and genre are also utilized in the experiments. Aggregation of the datasets, such as DIACU, is also beneficial as it brings together Church Slavonic texts along with other relevant texts, offering diachronic as well as geographical metadata that can be exploited in the experiments [18].

Sources are added to the data stream via three channels of acquisition, each of which has different integrity risks. Sources can be acquired as UD treebanks and added directly, retaining document identifiers, sentence identifiers, and token identifiers to ensure that evaluation downstream is aligned with annotation upstream [40, 32]. Sources can also be acquired as scholarly editions, which may be represented in an Extensible Markup Language (XML)/Text Encoding Initiative (TEI) format. TEI has advantages because it can retain structural information and editor interventions, however, TEI conversion does not solve the problem of encoding because non-standard characters and font-dependent code points are not altered and can cause problems downstream [92, 8, 50]. Finally, if portals do not offer a bulk download, then a form of structured downloading or scraping is used. In these cases, the text may be

legible on the portal but become non-interoperable after downloading if it uses PUA or other font-dependent systems [8, 50].

Figure 3.2 shows the behavior of the portal side of things, demonstrating how the text is displayed correctly in its controlled web environment despite its use of encoding methods that do not necessarily promote interoperability.

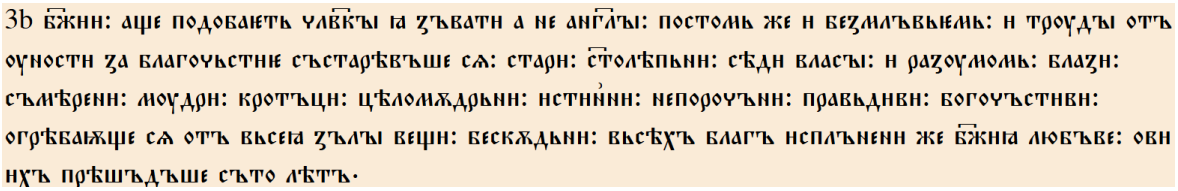

Figure 3.2: Portal-side rendering of Church Slavonic text in a controlled web environment (Cyrillomethodiana).

In all sources, the inclusion is controlled by licensing/access restrictions, availability of annotations, and representational stability. The resources with lemmas are given priority for training and main evaluation, while the resources with partial or no annotations are used for robustness evaluation and normalization validation. The resources with non-standard encodings, font-dependent representations or mixed diacritics are included if problematic code points can be reliably diagnosed and normalized according to documented procedures [8, 50, 15]. If metadata is provided, the period and region labels are preserved for controlled robust evaluation, taking into account diachronic and regional variation [18, 55].

3.2 Unified Representation, Cleaning, and Tokenization

All sources are then unified into a single representation that allows for consistent processing across formats and provides traceability for their sources. The aim of this step is to avoid information loss by instead making these differences explicit and traceable, while limiting the potential impact of extraction artifacts on the data.

Each text item is represented with a minimal set of metadata, including a source identifier, acquisition channel, language variant, and provenance information linking to repository references or portal URLs. If sources are aggregated is maintained at a dataset level and for a given period/region. If multiple titles exist, such as orthography, transliteration, and Latin, then these are maintained, and a single canonical identifier is used.

All sources are unified into a common internal representation with an explicit internal schema that includes levels such as Document with ID, provenance, and metadata, Segment, or paragraph/line blocks if available; Sentence, if available or reliably inferable; and Token, or surface form with optional lemma and morphosyntactic information. Sources already represented in UD format naturally map to this internal schema. Sources represented as TEI and web formats are segmented by using markup and layout information. Sentence segmentation is not performed on sources lacking strong evidence, such as editions of Church Slavonic text, as sentence boundaries can be arbitrarily defined and inconsistent across editions. Paragraph

or line blocks are maintained as a conservative representation since segmentation can have a significant impact on text data processing, particularly for historical sources.

Cleaning of the data is, by necessity, a conservative operation, focusing on the removal of extraction noise rather than the extraction of linguistic content itself. The text data obtained from the web often includes spurious whitespace, duplicate line breaks, and other boilerplate elements that arise during extraction. The text data obtained from the TEI may also include editorial markings that may be interpreted rather than blindly removed during cleaning. Therefore, the following three-step cleaning process is used, normalization of whitespace without the removal of meaningful separators, removal of obviously non-linguistic extraction noise, and retention of manuscript-relevant markings where plausible interaction with token boundaries is possible. Excessive normalization of the orthography is also avoided, since "non-standard" spellings often include diachronic or other relevant information. The UD treebank structure is, by necessity, defined by tokenization itself. For the TEI and web data, tokenization is performed after representational normalization, which avoids tokenization fragmentation caused by code-point inconsistencies. For data known to be space-delimited, whitespace-based tokenization is used as a fallback, with care taken over punctuation handling.

Subsequent to acquisition, cleaning, and standardization of encoding, the texts are converted into a form compatible with the UD format and the CoNLL-U format, retaining document provenance, `sent_id`, and token indices. This is for both model training and evaluation, as well as for the semi-automatic gold annotation process described in Chapter 4. The main driving force is to make sure that annotation and evaluation take place over fixed tokenization and compatible Unicode strings, rather than portal-specific encoding schemes.

3.3 Encoding Standardization

Historical Slavic texts have been collected via various heterogeneous digital channels, including web portals, legacy word processor documents, XML/TEI, and NLP-oriented formats like CoNLL-U. In many cases, the texts appear to render correctly but fail to achieve interoperability. A key factor in this failure has been the overuse of font-dependent practices and non-standard encoding schemes, especially Unicode PUA codes and mixed-script substitutions, which have been introduced for visual rather than linguistic reasons [8, 50, 15].

From an NLP standpoint, this is not a negligible problem. Tokenization, lexicography, and even character models assume that characters with the same visual representation have the same underlying encoding. However, in this scenario, characters with the same visual representation will have different underlying encodings (standard Unicode and PUA, respectively), resulting in a fragmented distribution, inflated lexis, and increased OOV error, even in the complete absence of actual linguistic variation. This has a number of negative consequences for dictionary-based models and neural network performance, in which incorrect relationships will be learned, depending on font conventions rather than actual valid morpho-

logical structure.

A common failure mode for this scenario is for the text to correctly render within the portal environment, owing to the font substitution, and then fail to render correctly in other tools, owing to PUA dependencies. This has been demonstrated in Figure 3.2, in which the portal environment correctly renders the text, including accurate shaping and diacritical marks. However, in the extracted text, PUA-dependent characters will often be missing or incorrect, resulting in poor readability and tokenization, as demonstrated in Figure 3.3.

Зb б□ж□□: аще подоба□ть □лв□кы ꙗ ꙁъват□ а не анг□лы: постомь же □ беꙁмлъвь□мь: □ троудъ отъ оуност□ ꙁа благо□ьст□□ състарѣвъше сѧ: стар□: с□толѣпьн□: сѣд□ власъ: □ раꙁоумомь: блаꙁ□: съмѣрен□: моудр□: кротъц□: цѣломѫдрьн□: □ст□нꙗ̀н□: непоро□ън□: правьд□в□: бого□ъст□в□: огрьбаѭще сѧ отъ вьсеꙗ ꙁълы вещ□: бескѫдьн□: вьсѣхъ благъ □сплънен□ же б□ж□ꙗ любъве: ов□ □хъ прѣшъдъше съто лѣтъ·

Figure 3.3: Text after extraction into standard tools, portal-dependent (PUA) characters degrade or disappear, fragmenting token identity.

Another issue arises in relation to this one in the process of format conversion. In other words, even after conversion to TEI format, if any non-standard code points exist in the text, they may be preserved in their original form during conversion to TEI format. As a result, TEI documents may be partially unreadable or unsearchable within XML editors.

Standardization starts with the processing of text as an ordered sequence of code points. In this respect, each character is checked for inclusion in the Unicode PUA, including the BMP PUA (U+E000–U+F8FF), the Supplementary PUA-A (U+F0000–U+FFFFD), and the Supplementary PUA-B (U+100000–U+10FFFD) [93, 8]. This diagnostic will track the number of distinct PUA code points, their counts, and their origins by source. At the same time, character sets are computed for each source and each corpus snapshot to determine the sources that provide the most representational noise, so that the most frequent PUA character sets can be mapped first.

The formulation of the PUA normalization problem as a mapping problem under uncertainty is motivated by the fact that the PUA code points have no inherent meaning. Instead, the intended character identity of the PUA code points needs to be reconstructed using evidence from the outside world. The mapping is informed by the recommendations for the standardization of the Unicode for Church Slavonic, with special attention given to the Unicode Technical Note #41 and related standardization efforts [8, 15]. The correctness of the mapping is checked against the Unicode charts and the documentation of the individual code points [93, 94]. The mapping tables are considered specific to the given corpus snapshot, taking into account the possibility that the portal encodings could change over time.

Representative examples of this mapping are shown in Table 3.1, which illustrates how observed PUA glyphs and code points are linked to their intended Cyrillic characters and standard Unicode equivalents.

Normalization is achieved by a deterministic transformation of an input string T into an

Table 3.1: Some observed PUA glyphs and code points mapped to the intended Cyrillic character and standard Unicode code point.

PUA character	PUA code point	Normalized character	Unicode code point
□	U+E205	и	U+0438
□	U+E012	Ѣ	U+0462
□	U+E20D	Ч	U+0427
□	U+E201	Ꙗ	U+0464
□	U+E342	◌҃	U+0487
□	U+E343	◌҈	U+0488
□	U+E20C	Н	U+041D
□	U+E018	ъ	U+044A
□	U+E20E	Ю	U+042E
□	U+E213	Ꙗ	U+0464

3b б҃жии: аще подобаѥть члвк҃ъı ꙗ ꙁъвати а не анг҃лъı: постомь же и беꙁмлъвьıѥмь: и троудъı отъ оуности ꙁа благочьстиѥ състарѣвъше сѧ: стари: с҃толѣпьни: сѣди власъı: и раꙁоумомь: блаꙁи: съмѣрени: моудри: кротъци: цѣломѫдрьни: истин҃ни: непорочъни: правьдиви: богочъстиви: огрѣбаѭще сѧ отъ вьсеꙗ ꙁълъı вещи: бескѫдьни: вьсѣхъ благъ исплънени же б҃жиꙗ любъве: ови ихъ прѣшъдъше съто лѣтъ·

Figure 3.4: After normalization: standardized Unicode code points render consistently across tested environments, preserving text integrity.

output string T', wherein replacement rules are applied if and only if a character is a member of a PUA set. A dictionary-based replacement technique is used, wherein each key is a PUA character and each value is its standardized equivalent Unicode character. Unmapped characters are left unchanged but noted for possible review, thus enabling easy extension of the mapping without data loss.

Let $R = \{(c_i, c_i') \mid c_i \in \text{PUA},\ c_i' \in \text{Unicode}\}$ denote the remapping relation.

The converter applies:

$$f(T) = T' = \sum_{c \in T} \begin{cases} R(c) & \text{if } c \in \text{PUA} \\ c & \text{otherwise.} \end{cases} \tag{3.1}$$

After normalization, standardized Unicode code points render consistently across tested environments and stabilize token identity, as illustrated in Figure 3.4.

While the Unicode mapping is correct, the visual accuracy is still contingent upon the support provided by the font and rendering engines. The typography for Church Slavonic relies on specific glyphs and Open Type behavior to display diacritics, titla, and historical script in a way that is consistent with traditional practice [8, 50]. The normalized output is then validated with Unicode Church Slavonic fonts provided by the Slavonic Computing Initiative, including Ponomar, in order to verify that the standardized code points preserve the intended historical glyphs and diacritics, as illustrated in Figure 3.5 [7].

The quality of encoding-aware standardization can be evaluated intrinsically and extrinsi-

3b Б҃жии: аще подобаѥть чл҃вкꙑ ꙗ ѕъвати а не анг҃лꙑ: постомь же и беѕмлъвьѥмь: и троудꙑ отъ оуности ѕа благочьстиѥ състарѣвъше сѧ: стари: ѿтолѣпьни: сѣди власꙑ: и разоумомь: блаѕи: съмѣрени: моудри: кротъци: цѣломѫдрьни: истин̾ни: непорочъни: правьдиви: богочъстиви: огрѣбаѭще сѧ отъ вьсеꙗ ѕълꙑ вещи: бескѫдьни: вьсѣхъ благъ испълънени же Б҃жиꙗ любъве: ови ихъ прѣшьдъше съто лѣтъ·

Figure 3.5: OCS text rendered with the Ponomar Unicode font for visual validation of historical glyphs and diacritics.

cally in terms of statistics over corpora. Intrinsic evaluations include coverage over mappings, idempotence, and stabilization of character sets as indicated by reductions in unique PUA code points and mixed script artifacts. Extrinsic evaluations involve reductions in representational OOV inflation and improvements in stability of lexical lookups. All acquisition and preprocessing steps are versioned, with each corpus snapshot archived with mapping tables, diagnostics logs, and statistics to facilitate traceability and reproducibility between versions.

Chapter 4

Gold Resources for Early Slavic Lemmatization

This chapter will discuss the gold resources developed and crafted for Early Slavic lemmatization in this thesis. It will include details such as what is annotated and on which representation of the text, how the annotations are encoded to ensure interoperability with the two main ecosystems for historical language processing (UD pipelines and TEI-based philology), and what decision procedures are followed to ensure consistency of lemma labels across spelling variants and other forms of ambiguity. The chapter will also discuss the proposed workflow for semi-automatic annotation, with a focus on how it is tailored for efficient use under time and resource constraints, and finally, the statistics, splitting, and validation procedures applied to guarantee repeatability and linguistic quality of the gold standard material.

4.1 Annotation Targets and Scope

The primary focus of annotation in this thesis is the lemma for each token, which is a canonical form that all inflected forms map to, helping with indexing, concordance, and other morpho-syntactic analyses. This is especially important for Early Slavic, as it has limited supervision, changes in spelling, and spelling rules that increase data sparsity for models that focus on token-level data.

The gold standard resources focus on the Early Slavic languages included in the experimental workflow, with emphasis on OCS and OES as they appear in UD. Where applicable, they have been supplemented by other digitized editions that have been introduced throughout the construction and standardization of the data, as described in Chapter 3. In all cases, priority is given to datasets that have stable identifiers and token-level segmentation, which is important for lemmatization since there needs to be a traceable link between token boundaries and lemma labels.

The tokens are defined after the acquisition, cleaning, and normalization of the encoding, as discussed in Chapter 3. In the case of sources that are compatible with UD, the tokenization is naturally in compliance with the UD tokenization conventions. In the case of sources

that are not compatible with UD, whitespace tokenization is used, with punctuation tokens represented when present and informative for the alignment. In the case of sources that have little or no punctuation or use manuscript breathing marks, no additional tokenization is added beyond that which is necessary for the subsequent use cases, with a clear mapping provided from the original to the tokenized form.

The gold lemma annotation is anchored to the normalized form of the tokens. In other words, the labeled token strings are the output of the PUA to Unicode normalization layer, as discussed in Chapter 3. This ensures that the lemma annotation is not affected by encoding issues but is focused on linguistic variation. At the same time, the policy does not discard historically significant graphemic variation, such as redactional or editorial variation, unless it is purely representational, such as a PUA character instead of its standard Unicode equivalent.

Although lemmatization is the focus of this thesis, the annotation scheme is designed to remain interoperable with gold resources that include morphosyntactic annotation. In UD treebanks, lemma annotations are linked to additional token-level information, including UPOS, XPOS, FEATS, and dependency relations. This information supports ambiguity resolution and the construction of baseline models. For non-UD formats, lemma annotations are maintained at the token level, while optional POS and morphological features are preserved when available, allowing the corpus to be enriched or converted to other formats in future work.

4.2 Annotation Format and Interoperability (UD/TEI)

The gold resources in historical NLP should remain accessible across two environments, NLP pipelines and tests, where UD-style tokenization is the norm, and philological workflows, where TEI/XML is often used due to its ability to represent document structure, editorial choices, and metadata. This thesis supports a single internal representation that can be serialized to both CoNLL-U and TEI-style outputs.

If the data source is a UD treebank, the gold signal is taken to be the `LEMMA` field in the CoNLL-U file. UD has the advantage of providing a fixed token position in a sentence, a clear separation between `FORM` (surface form) and `LEMMA`, and optional morphosyntactic context (`UPOS` and `FEATS`), which is useful for analytical and baseline approaches (e.g., wordform dictionary lookup conditioned on POS/feature constraints). For interoperability and traceability, sentence IDs and document IDs are copied whenever available, and preprocessing operations are restricted to avoid interfering with token position. Standardization of encoding, e.g., PUA to Unicode, is done in a way that respects token position and is documented whenever applied.

In the case of TEI/XML source materials, or their equivalent after conversion into TEI format, the lemma is encoded as an attribute at the token level, which is appropriate for standard tokenization in TEI. In cases where tokenized TEI is available, the lemma is stored on the `<w>` elements, (or their equivalent token-level elements), as a dedicated attribute, e.g., lemma="...". In cases where source materials were not tokenized, decisions regarding tok-

enization are not encoded within the TEI transcription level; instead, a derived token level is maintained, which is stand-off where possible, so that philological structure is preserved, and token/lemma levels, which are appropriate for NLP, remain revisable and reproducible.

One effect of this architecture is that UD-oriented experimentation and TEI-oriented philological inspection can be performed simultaneously without duplication of resources. The internal schema will, at least, hold the normalized token text, the label for the lemma, optional POS/features, stable token/sentence/document identifiers, and provenance metadata, such as source corpus, document, and edition.

4.3 Annotation Guidelines and Decision Rules

The main difficulty in Early Slavic lemmatization is not only the rich morphological system but also the potential ambiguity resulting from orthography, editorial choices, and differing resource conventions. To ensure that gold lemma choices are consistent across sources and annotation passes, this work includes explicit decision procedures that distinguish between representational normalization and linguistic normalization and document the procedures for determining the canonical lemma strings.

The lemma strings conform, where possible, to the citation standards of the underlying scholarly resource. For UD sources, existing lemma choices are adopted by default unless there is evidence of clear encoding artifacts or systematic inconsistencies. For newly curated or corrected items, the lemma choices aim to match the dictionary headwords used in the appropriate tradition (OCS/OES lexicography), be stable across predictable orthographic variation, and avoid encoding-based artifacts such as PUA characters or inconsistent diacritic composition.

Diacritics and combining marks are represented in normalized Unicode following conversion. Diacritics are preserved when they represent the intended orthography used within the resource; they are not used to represent differences in typography. This aligns the lemma with linguistic identity rather than typography or display identity.

Ambiguity is treated as the primary issue for surface forms that map to multiple lemma choices; the gold lemma is chosen based on the sentential context in the immediate vicinity of the word and, when available, the morphosyntactic tags. This is based on expert procedures for handling ambiguity and overcomes the significant limitation of dictionary lookup based solely on surface form, which fails systematically for minority sense or grammatical usage of ambiguous words. Proper names are lemmatized to the canonical name representation within the source tradition. When there are multiple orthographic variants for the same proper name, a canonical lemma is chosen from the variants following normalization, with the mapping noted when appropriate in internal notes.

The handling of non-lexical tokens is also conservative in nature, ensuring compatibility with standard evaluation approaches. Punctuation tokens are handled conservatively in the lemma field; in CoNLL-U, "_" is used only when the lemma is unspecified, while punctuation

marks may otherwise retain their punctuation symbol as the lemma (or an empty lemma attribute in TEI format) unless a target framework is specified for the evaluation task. Editorial markings, such as uncertain readings or expansion brackets, are standardized or eliminated during the preprocessing step described in Chapter 3. If they occur as tokens, they are ignored during the evaluation of the lemma to avoid confusion between editorial markings and lexical content of the word itself.

Numerical forms and abbreviations conform to the conventions of the source material. Numerical tokens are either excluded from evaluation or have a stable lemma representation (e.g., normalized numeric string), depending on the target protocol for the task. For abbreviations, their handling is also standardized, ensuring that when the surface transcription reflects the expanded abbreviation, the lemma also corresponds to the expanded lexical item. When abbreviation marks are preserved within the transcription, the lemma corresponds to the intended lexical item while ignoring the abbreviation marks, treating them only as graphic information rather than contributing to the lemma string itself.

For unattested or uncertain forms, the best-supported lemma is assigned on the basis of morphology and context rather than omitting the token from evaluation. This also tests the robustness of the model, especially its ability to generalize to OOV forms.

4.3.1 Semi-Automatic Annotation Workflow

The gold-standard annotation process is costly, particularly because it involves language expertise along with an understanding of editorial practices and orthographic variation. Following the trend that has been noted in low-resource NLP [73], where time spent annotating is a major bottleneck, we use a semi-automatic process. This minimizes the repetitive work while still allowing expertise to be used when necessary.

We start by collecting all available token-level lemma annotations from existing resources in Early Slavic, with a primary focus on UD treebanks and other compatible curated resources. These annotations are used in two ways: (i) to train the neural lemmatizer architecture discussed in Chapter 5, and (ii) to derive a dictionary lookup table that maps surface forms to lemmas, yielding a high-precision signal for frequent types.

For the texts that need new annotations, we apply the encoding standardization layer (Chapter 3) and then proceed with converting the standardized text into CoNLL-U format, while keeping stable sentence IDs and token positions. This ensures that the output of the annotation layer is directly usable for UD-style training and evaluation, and it makes it easier to add more layers of annotations (such as POS and morphological features) without having to access the original corpus. Figure 4.1 summarizes the semi-automatic annotation procedure adopted in this thesis, from annotation-ready text to the final verified dataset.

Given a tokenized CoNLL-U file, we derive the lemma candidates from two different but complementary sources. First, a dictionary-based lookup is carried out using the aggregated lemma inventory. When a surface form is part of the training inventory, its associated lemmas along with their counts are obtained. Second, the neural lemmatizer is used to derive a

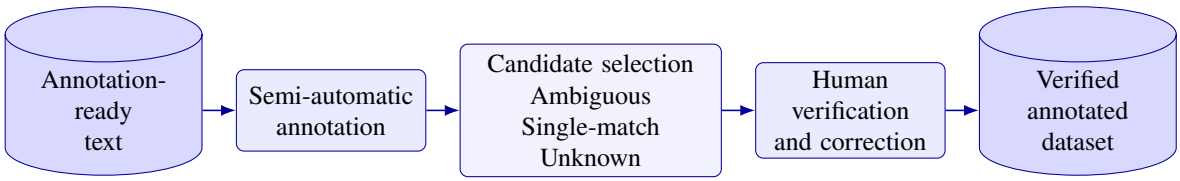

Figure 4.1: Semi-automatic annotation workflow used to construct the verified lemma-annotated dataset. Starting from annotation-ready text, the process combines automatic candidate generation with expert verification and correction.

prediction for each token, including unseen token types, by applying the model.

Next, the tokens are categorized into three queues of annotations, which are optimized to minimize expert effort:

1. **Unambiguous seen types:** tokens whose surface form maps to exactly one lemma in the aggregated inventory. These cases are assigned automatically from the inventory.

2. **Ambiguous seen types:** tokens whose surface form maps to multiple lemmas. For these cases, we provide a ranked list of up to five candidates, ordered by their frequency in the aggregated inventory, together with the neural model prediction as an additional suggestion.

3. **Unseen (OOV) types:** tokens whose surface form does not occur in the aggregated inventory. These cases rely on the neural model prediction as the primary candidate and are flagged for manual review.

This limits the automatic decision-making to high-confidence cases, while explicitly highlighting the presence of ambiguity and OOV behavior, which is an important aspect of the orthographically varied historical corpora.

The suggested lemma candidates are validated and corrected by the human expert annotator. The file is designed in such a manner that the review process can be accomplished quickly, considering that the suggested list of token types is clustered, along with the number of cases for each token type. Changes to the lemma decision are made by human experts, followed by the regeneration of the suggested list to ensure that the process does not involve repeated human intervention for the same cases.

The final output is the human-validated CoNLL-U dataset, which contains permanent IDs and normalized Unicode strings. This serves as the gold standard, which is used for the experiments in the subsequent chapters.

4.4 Gold Dataset Statistics and Splits

The gold resources are accompanied by descriptive statistics, which make it easier to compare across experiments, quantify the amount of representational noise removed by standardization, and provide evidence of the effectiveness of this process. Specifically, the current

annotated dataset consists of 29,678 tokens drawn from multiple OCS source groups. In addition to its overall size, the dataset exhibits substantial lexical diversity; it contains 25,517 unique word forms and 23,803 unique lemmas. Beyond the annotated subset used in the present experiments, the broader source collection is substantially larger. It includes more than 200 texts comprising approximately more than 5 million tokens, and may support future annotation efforts.

Table 4.1 reports approximate token counts for the annotated subset used for lemma annotation and validation. These statistics are provided after normalization to ensure that any improvements in coverage or downstream accuracy are not artifacts of encoding mismatches. This also enables quantifying the reduction of representational OOV, which serves as a necessary infrastructure rather than a model improvement.

Table 4.1: Current source composition of the annotated OCS material.

Source group	**Origin**	**Tokens**
Cyrillomethodiana	Cyrillomethodiana portal	5,789
Creed	Pravmir: *Simvol very*	77
Dictionary texts	Azbyka	23,812
Total		29,678

We use data splits to assess both in-domain accuracy and out-of-domain robustness, including shifts across historical domains. When available, source splits are used to assess out-of-domain generalization across time, edition, or portal. Finally, OOV-focused splits are provided to ensure a proportion of the test set consists of OOV tokens, providing a direct assessment of morphological generalization, especially under OOV-heavy conditions, which are particularly prevalent in the Early Slavic data sets. When using UD resources, the pre-existing train/dev/test splits are used when available.

4.5 Quality Control and Validation

The value of gold resources is only realized when they display consistency, encoding integrity, and reproducibility. Hence, validation is performed on four levels, namely, structural integrity, encoding integrity, lemma-field integrity, and linguistic consistency.

Structural validation checks ensure proper token index and identifier form (no missing or duplicate token IDs), sentence boundaries, and document identifier uniqueness. Encoding validation checks ensure no PUA code points remain after standardization if not documented, uniform normalization form is applied throughout the files (consistency of the combining strategy), and normalization is idempotent (reapplication of the converter does not change the result).

The integrity of lemma fields is verified, ensuring lemma fields are present for all lexical tokens in the gold resources and explicit for absent values if applicable. For UD-derived resources, this includes verifying non-empty values of the LEMMA field when expected and ensuring token alignment is preserved and has not been disrupted by preprocessing. For

TEI-derived resources, this includes verifying lemma attributes are correctly attached to the appropriate token layer and remain synchronized with exported token sequences.

Linguistic consistency checks are performed on the corpus as a whole and are used to detect outliers, which are likely to reflect either true linguistic ambiguity or hidden preprocessing bugs. One-to-many mappings, where a word type maps to many lemmas (expected for ambiguous forms, questionable for extreme cases), many-to-one mappings, where many word types map to a single lemma (expected for inflectional variation, questionable if driven by encoding issues), and improbable lemma string detection (stray markup, numbers, or unexplained combining) are all included in linguistic consistency checks. Systematic incompatibility of lemma choices with other available information as a check, with corrections to the gold data or preprocessing rules as needed, with all changes tracked via snapshots of the corpus at particular versions.

Chapter 5

Neural Lemmatization: OldSlavicLemma

This chapter presents OldSlavicLemma, our neural, dictionary-free lemmatization system for Early Slavic languages. The rationale is that we can treat lemmatization as a character-level sequence transduction problem. This problem can be solved by a model. conditioned on a small context window around the input word. This is motivated by three factors: (i) the complex inflectional morphology of OCS and OES, (ii) the presence of orthographic and editorial variation, and (iii) lexical ambiguity where words can have different lemmas depending on context. Unlike earlier hybrid approaches that combine dictionary information with neural networks, such as Stanza [78, 79], or edit rules, such as UDPipe [89, 90], our model is purely neural and can generalize beyond its training data. The implementation is released on GitHub[1].

5.1 Problem Formulation

We formulate lemmatization as a sequence transduction task defined at the character level. The model receives as input a context window that includes both the token to be lemmatized and its sentential context. Let the context window S of size K be defined as:

$$S = (w_{-K}, \ldots, w_{-1}, w_0, w_{+1}, \ldots, w_{+K}), \tag{5.1}$$

where w_0 denotes the token whose lemma is to be predicted, and $w_{-K}, \ldots, w_{-1}, w_{+1}, \ldots, w_{+K}$ denote the contextual tokens on its left and right sides. The hyperparameter K controls the amount of contextual information made available to the model. In our implementation, we set $K = 2$, so that the model observes two tokens on each side of the target within S.

The context window is converted into a serialized input stream $\mathbf{x}$ in which each token w_i is represented as a sequence of characters drawn from the finite alphabet C:

$$\mathbf{x} = \langle \text{sos} \rangle \ \text{chars}(w_{-K}, \ldots, w_{-1}) \perp \text{chars}(w_0) \perp \text{chars}(w_{+1}, \ldots, w_{+K}) \ \langle \text{eos} \rangle.$$

The stream $\mathbf{x}$ is formed by concatenating character sequences. The function $\text{chars}(\cdot)$ maps a

[1] https://github.com/usmannawaz01/EarlySlavicLemmatization

token to the ordered sequence of its characters. The special symbol ⊥ marks the boundaries of the target token relative to the left and right context, whereas ⟨sos⟩ and ⟨eos⟩ indicate the start and end of **x**. In addition, ⟨pad⟩ is used for padding, and ⟨unk⟩ represents unknown or out-of-vocabulary characters. It should be noted that the alphabet C includes all characters observed in the training data, including letters, diacritics, and punctuation, some of which may carry language-specific information. For example, in OCS texts the diacritical mark titlo indicates omitted letters or abbreviations, especially in words referring to holy persons or religious concepts. This formulation of the input **x**, allows the model to identify which segment of the sequence corresponds to the token of interest while preserving the surrounding context.

Given the serialized input **x**, the model produces as output a character sequence representing the predicted lemma of the target token w_0:

$$y = (y_1, y_2, \ldots, y_T), \qquad y_t \in C \quad \forall t, \tag{5.2}$$

where T is the length of the target character sequence, each y_t denotes the t-th character of the canonical lemma, and the lemma is generated character by character sequentially from the same alphabet C.

Training is carried out under a conditional character-level language-modeling objective, where the probability of the lemma sequence is maximized given the serialized input stream:

$$p(y \mid \mathbf{x}) = \prod_{t=1}^{T} p(y_t | y_{<t}, \mathbf{x}), \tag{5.3}$$

where $y_{<t} = (y_1, \ldots, y_{t-1})$ denotes the prefix of characters generated before step t. Equation (5.3) defines an autoregressive factorization in which each lemma character y_t is predicted from two sources: (i) the serialized input stream **x** (Equation (5.1)), and (ii) the previously generated output prefix $y_{<t}$. Under this formulation, the lemmatization problem becomes a character-level Seq2Seq task.

5.2 Model Architecture

Our proposed OldSlavicLemma adopts a character-level encoder–decoder architecture as illustrated in Fig. 5.1. The encoder processes embeddings derived from the serialized input stream **x**, which contains the character sequence of the target token together with the characters of its left and right context (K tokens on each side). The decoder then generates the lemma ℓ autoregressively, producing one character at a time.

The encoder consists of two stacked BiLSTM layers, which progressively transform the input into increasingly informative character-level representations. The first BiLSTM is intended to capture fine-grained orthographic and sequential regularities, while the second layer builds higher-order contextual abstractions over those representations. Between these two recurrent layers, we employ a cross-layer multi-head attention (CLMA) mechanism that ap-

plies attention over BiLSTM outputs [54] using the standard Q–K–V formulation of scaled dot-product attention [97].

In this module, the output of the first BiLSTM is used as the source of queries (Q), whereas the output of the second BiLSTM is used as the source of keys and values (K, V). The effect of this design is to re-interpret lower-level character representations through the lens of higher-level contextual states, thereby combining detailed morphological evidence with broader contextual structure. Unlike standard self-attention, which models relations among positions within the same layer, our cross-layer formulation promotes inter-layer information flow and tighter integration of representations.

Lemma generation is handled by a unidirectional LSTM decoder [43] operating in an autoregressive manner [37]. At each decoding step, the decoder hidden state acts as the query (Q), while the encoder representation provides the keys and values (K, V) through an encoder-decoder cross-attention (EDCA) mechanism. This design allows every decoding step to remain conditioned on the full encoded input sequence rather than depending only on local or position-aligned slices. As a result, the model can better resolve ambiguous forms and can generalize to unseen surface realizations by dynamically retrieving the most relevant encoded information during generation.

In the following, we present the components of the architecture in detail.

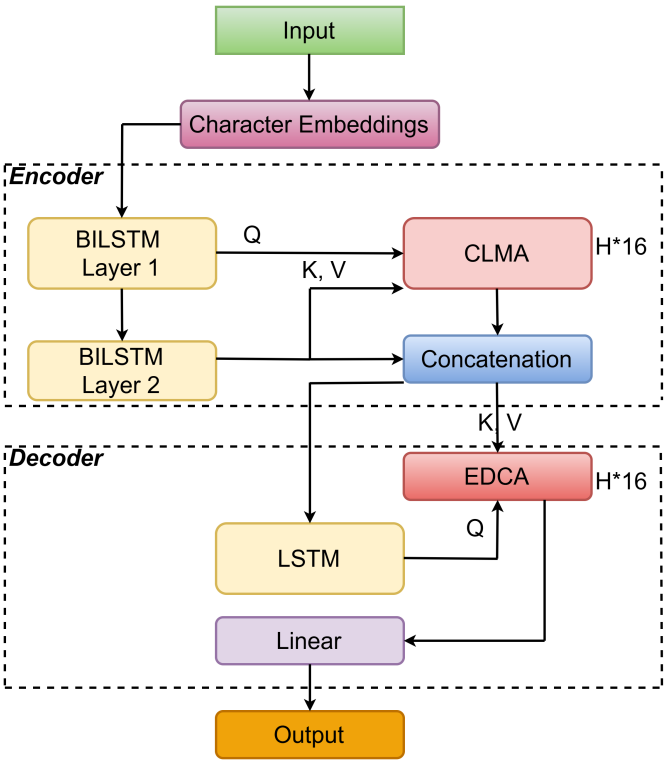

Figure 5.1: Architecture of our proposed encoder–decoder lemmatization model. Two stacked BiLSTMs are connected by cross-layer multi-head attention (CLMA), while the decoder employs cross-attention over encoder states (EDCA).

5.2.1 Character Embedding Layer

The architecture begins by transforming discrete character symbols to continuous vector representations suitable for neural processing. Let C denote the finite character vocabulary observed from the training corpus, including, as noted above, diacritics, punctuation, the separator symbol $\perp$, and the four reserved tokens ($\langle$sos$\rangle$, $\langle$eos$\rangle$, $\langle$pad$\rangle$, $\langle$unk$\rangle$). We define a trainable embedding matrix:

$$\mathbf{E} \in \mathbb{R}^{|C| \times d_c} \tag{5.4}$$

where $|C|$ is the vocabulary size and d_c is the embedding dimension. Each row of $\mathbf{E}$ corresponds to the vector representation of one character in C.

For an input character sequence

$$\mathbf{x} = (x_1, x_2, \ldots, x_N), \quad x_n \in C \tag{5.5}$$

where N denotes the sequence length and x_n is the n-th character, embedding lookup maps every symbol into its dense representation:

$$\mathbf{X} = \left[\mathbf{E}[x_1]; \mathbf{E}[x_2]; \ldots; \mathbf{E}[x_N]\right] \in \mathbb{R}^{N \times d_c} \tag{5.6}$$

Here, $\mathbf{E}[x_n] \in \mathbb{R}^{1 \times d_c}$ denotes the embedding vector of the n-th character; the operator $(;)$ denotes vertical concatenation, and $\mathbf{X}$ is the resulting embedded sequence passed to the encoder. In our implementation, the trainable embedding weights $E_{i,j}$ are initialized by sampling from a standard normal distribution with zero mean and unit variance.

The embedding associated with the padding token $\langle$pad$\rangle$ is constrained to remain zero:

$$\mathbf{E}[\langle \text{pad} \rangle] = \mathbf{0} \in \mathbb{R}^{d_c}. \tag{5.7}$$

Equation (5.7) ensures that padding does not contribute to either gradient updates or to subsequent computations. Through this layer, the model acquires distributed representations over characters, enabling it to capture regularities in orthography, including diacritic usage and variant spellings, while also treating the separator symbol $\perp$ as an explicit structural indicator of target boundaries.

5.2.2 Encoder

The embedded character sequence $\mathbf{X} \in \mathbb{R}^{N \times d_c}$ (Sec. 5.2.1), where N denotes the sequence length and d_c the embedding dimension, is processed by two stacked BiLSTM layers. The first layer is intended to model local orthographic and morphological patterns, and the second layer further refines these representations. Each LSTM unit follows the standard formulation, with gating mechanisms and a tanh nonlinearity for both candidate states and hidden outputs. Let H denote the hidden size in each direction. The recurrence of the first BiLSTM is defined

as:

$$\begin{aligned}\overrightarrow{\mathbf{h}}_t^{(1)} &= \mathrm{LSTM}_{\rightarrow}^{(1)}(\mathbf{E}[x_t], \overrightarrow{\mathbf{h}}_{t-1}^{(1)}),\\ \overleftarrow{\mathbf{h}}_t^{(1)} &= \mathrm{LSTM}_{\leftarrow}^{(1)}(\mathbf{E}[x_t], \overleftarrow{\mathbf{h}}_{t+1}^{(1)}).\end{aligned}$$

Here, $\mathbf{E}[x_t] \in \mathbb{R}^{d_c}$ is the embedding associated with the t-th character, $\overrightarrow{\mathbf{h}}_t^{(1)} \in \mathbb{R}^H$ is the forward hidden state, and $\overleftarrow{\mathbf{h}}_t^{(1)} \in \mathbb{R}^H$ is the backward hidden state. Their concatenation is defined as:

$$\mathbf{H}_t^{(1)} = [\overrightarrow{\mathbf{h}}_t^{(1)}, \overleftarrow{\mathbf{h}}_t^{(1)}] \in \mathbb{R}^{2H} \quad \forall t \tag{5.8}$$

and $\mathbf{H}^{(1)} \in \mathbb{R}^{N\times 2H}$ denotes the sequence of context-sensitive states produced by the first layer. These representations are then consumed by the second BiLSTM, which computes:

$$\begin{aligned}\overrightarrow{\mathbf{h}}_t^{(2)} &= \mathrm{LSTM}_{\rightarrow}^{(2)}(\mathbf{H}_t^{(1)}, \overrightarrow{\mathbf{h}}_{t-1}^{(2)}),\\ \overleftarrow{\mathbf{h}}_t^{(2)} &= \mathrm{LSTM}_{\leftarrow}^{(2)}(\mathbf{H}_t^{(1)}, \overleftarrow{\mathbf{h}}_{t+1}^{(2)}).\end{aligned}$$

Again, for each time step t, the output $\mathbf{H}_t^{(2)}$ is formed by concatenating the hidden states from the forward and backward directions, and $\mathbf{H}^{(2)} \in \mathbb{R}^{N\times 2H}$ denotes the higher-level contextualized sequence resulting from Eq. (5.9).

$$\mathbf{H}_t^{(2)} = [\overrightarrow{\mathbf{h}}_t^{(2)}; \overleftarrow{\mathbf{h}}_t^{(2)}] \in \mathbb{R}^{2H}. \tag{5.9}$$

Although BiLSTMs are effective at capturing sequential dependencies, their ability to directly relate distant positions remains challenging. To address this, we employ a CLMA block in which the outputs $\mathbf{H}^{(1)}$ of the first BiLSTM serve as queries, while the outputs $\mathbf{H}^{(2)}$ of the second BiLSTM serve as keys and values for a multi-head attention mechanism [97]. This design explicitly links lower-level representations to higher-level contextual abstractions. The corresponding linear projections are defined as:

$$\begin{aligned}\mathbf{Q} &= \mathbf{H}^{(1)}\mathbf{W}^Q + \mathbf{b}^Q,\\ \mathbf{K} &= \mathbf{H}^{(2)}\mathbf{W}^K + \mathbf{b}^K,\\ \mathbf{V} &= \mathbf{H}^{(2)}\mathbf{W}^V + \mathbf{b}^V\end{aligned}$$

where $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V$ are trainable projection matrices and $\mathbf{b}^Q, \mathbf{b}^K, \mathbf{b}^V$ are bias vectors. After linear projection, $\mathbf{Q}, \mathbf{K} \in \mathbb{R}^{N\times d_k}$ and $\mathbf{V} \in \mathbb{R}^{N\times d_v}$.

Within the multi-head attention layer, $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ are partitioned into h heads, each with dimensionality d_k/h and d_v/h. Scaled dot-product attention, as given in Equation (5.10), is applied independently to each head, after which the resulting outputs $\mathbf{A_h}$ are concatenated and projected back. Specifically, scaled dot-product attention is computed as:

$$\mathbf{A}_h = \mathrm{softmax}\left(\frac{\mathbf{Q}_h\mathbf{K}_h^\top}{\sqrt{d_k}}\right)\mathbf{V}_h. \tag{5.10}$$

The attention mechanism computes similarity by taking dot products between a query and all keys, scaling the result by $\sqrt{d_k}$, and applying softmax to obtain normalized weights reflecting the relative relevance of positions for the output computation. These weights are then used to construct a weighted combination of the value vectors. The factor $1/\sqrt{d_k}$ is used to prevent the logit magnitude from growing, which can lead to unstable training and slow convergence.

Finally, the outputs $\mathbf{H}^{(2)}$ of the second BiLSTM are concatenated with the outputs of the CLMA layer to obtain the encoder features $\tilde{\mathbf{E}}^{\star} \in \mathbb{R}^{N\times 4H}$.

5.2.3 Decoder

The decoder generates lemma characters autoregressively under *teacher forcing* [104]. Given the gold prefix $\mathbf{y}_{1:L-1}$, we build per-step inputs by concatenating lemma embeddings with encoder features:

$$\mathbf{D} \;=\; \left[\, \mathbf{E}_y(\mathbf{y}_{1:L-1}) \;;\; \tilde{\mathbf{E}}^{\star}_{1:L-1} \,\right] \;\in\; \mathbb{R}^{(L-1)\times(d_y+4H)}.$$

$\mathbf{E}_y(\mathbf{y}_{1:L-1}) \in \mathbb{R}^{(L-1)\times d_y}$ are the decoder input embeddings with d_y indicating the embedding dimension. During training, the decoder input embeddings $\mathbf{E}_y(\mathbf{y}_{1:L-1})$ are obtained by passing the gold lemma characters (under teacher forcing) through the shared character embedding layer used by the encoder. $\tilde{\mathbf{E}}^{\star}_{1:L-1} \in \mathbb{R}^{(L-1)\times 4H}$ are derived from the encoder output, length-matched to $L-1$ by truncation or padding.

A unidirectional LSTM consumes $\mathbf{D}$ and produces decoder states:

$$\mathbf{S} \;=\; \mathrm{LSTM}_{\mathrm{dec}}(\mathbf{D}) \;\in\; \mathbb{R}^{(L-1)\times 2H}.$$

Here, $\mathbf{S}_t \in \mathbb{R}^{2H}$ is the decoder hidden state at step t. Each LSTM unit applies gating mechanisms to regulate information flow, producing context-sensitive hidden states that capture both short- and long-range dependencies.

In the encoder–decoder cross attention (EDCA) layer, we employs *multi-head cross-attention*, where multiple attention heads operate in parallel. The decoder's hidden states act as queries, while the encoder representations provide keys and values. This mechanism enables each decoding step to selectively attend to all input positions, closely following the classical Seq2Seq attention formulations [91, 10, 59]. The Transformer architecture [97] adopts the same structure, queries from the decoder and keys/values from the encoder but extends it through multi-head attention, offering a unified and more expressive formulation of attention.

$$
\begin{aligned}
\mathbf{Q}^{\text{dec}} &= \mathbf{S}\mathbf{W}^{Q} + \mathbf{b}^{Q}, \\
\mathbf{K}^{\star} &= \mathbf{E}^{\star}\mathbf{W}^{K} + \mathbf{b}^{K}, \\
\mathbf{V}^{\star} &= \mathbf{E}^{\star}\mathbf{W}^{V} + \mathbf{b}^{V}.
\end{aligned}
$$

Scaled dot-product cross-attention is computed as:

$$
\mathbf{C} = \text{softmax}\left(\frac{\mathbf{Q}^{\text{dec}}\mathbf{K}^{\star\top}}{\sqrt{d_k}}\right)\mathbf{V}^{\star}.
$$

Finally, logits and probabilities for the next-character $\mathcal{Y}$ are:

$$
\begin{aligned}
\mathbf{z}_t &= \mathbf{W}_{\text{out}}\mathbf{C}_t + \mathbf{b}_{\text{out}} \\
p(y_t \mid y_{<t}, \mathbf{x}) &= \text{softmax}(\mathbf{z}_t).
\end{aligned}
$$

At step t, the decoder takes the cross-attention output $\mathbf{C}_t$ and projects it through a linear transformation to obtain logits $\mathbf{z}_t$. Each entry of $\mathbf{z}_t$ assigns a score to a candidate next character, and the subsequent softmax operation converts these scores into a probability distribution over the next character.

5.2.4 Training Objective and Optimization

The training objective is the negative log-likelihood of the gold lemma characters. For a mini-batch of B examples, where the b-th example has lemma length T_b, we minimize the cross-entropy loss

$$
\mathcal{L}(\theta) = \frac{1}{N_{\text{char}}} \sum_{b=1}^{B} \sum_{t=1}^{T_b} \sum_{c \in C} -y^{*(b)}_{t,c} \, \log p_\theta\left(c \mid y^{(b)}_{<t}, x^{(b)}\right), \tag{5.11}
$$

where p_θ denotes the model's predictive distribution, $y^{(b)}_{<t}$ is the gold lemma prefix at step t, and $y^{*(b)}_{t,c}$ is the one-hot indicator for the gold character and $N_{\text{char}} = \sum_{b=1}^{B} T_b$ is the total number of non-padding target characters in the batch. This loss yields an average per-character loss. It prevents longer lemmas from dominating the objective and keeps the loss scale consistent across batches with variable lengths.

Dropout is applied in the encoder, attention, and decoder components. For optimization, we use the Lion optimizer [20], together with a learning-rate scheduler and early stopping.

Table 5.1 reports the hyperparameters used in model implementation and training. Optimization proceeds through mini-batch updates. We also apply gradient clipping and replace input characters with the `<unk>` token with probability 0.05, a strategy intended to improve robustness to spelling variation.

Table 5.1: Hyperparameters used in all experiments.

Batch size	32
Epochs	75
Learning rate	1.5×10^{-4}
Embedding Dimension (d_c , d_y)	96
Latent Dimension (H)	128
Attention heads (h)	16

5.3 Reproducibility and integration with the thesis pipeline

OldSlavicLemma is evaluated under the same representation-aware pipeline used for the baseline system and the corpus resources described earlier (Chapter 3). In particular, the encoder consumes normalized Unicode strings produced by the PUA-to-Unicode standardization layer so that model capacity is not wasted on learning portal-specific or font-specific encoding artifacts. Keeping normalization constant across models ensures that improvements can be attributed to the lemmatizer rather than to hidden preprocessing drift, and it supports fair comparisons with dictionary baselines and UD pipelines. The released implementation includes the preprocessing steps required to construct context windows, serialize character streams with boundary markers, train the model, and run evaluation in a reproducible manner.

Chapter 6

Comparative Systems and Baseline

This chapter describes the underlying baselines and comparative approaches adopted throughout this thesis. There are two objectives. First, it uses strong baselines that have been extensively used in UD evaluations and historical NLP benchmarks. Therefore, the baselines have value beyond the specific snapshot of the dataset. Secondly, this chapter outlines the underlying inductive biases of the baselines: lexicon lookup, edit rule transformation, and neural sequence transduction. These biases play a significant role in the types of errors that dominate historical low-resource datasets, such as out-of-vocabulary words, spelling errors, and lemma ambiguity. In this chapter, the baseline are evaluated with the same scoring mechanisms as those discussed in Chapter 7.

The main baselines have been evaluated under gold tokenization to isolate lemmatization from any tokenization effects. In addition, for the Early Slavic UD v2.12 treebanks, we also include a raw-text-to-CoNLL-U comparison in which the Stanza tokenizer is used, and OldSlavicLemma is applied to the resulting tokenized files.

Under gold tokenization, results are reported as lemma accuracy; for raw end-to-end pipeline evaluation, we report the official Lemmas F1 score. We compare OldSlavicLemma against lemmatizers representing four design paradigms:

1. Edit-rule-based approaches, which learn transformation rules from surface forms to lemmas;

2. Hybrid approaches, which combine dictionary lookup with neural backoff for infrequent or unseen words;

3. Shared-task systems evaluated on historical benchmark data; and

4. A dictionary-based baseline, which serves as a transparent memorization-oriented comparator.

6.1 Dictionary-Based System

A simple dictionary-based lemmatizer provides a clear and informative baseline. Despite its simplicity, the approach may remain competitive in a setting with a high degree of repetition at the surface form level and a consistent mapping to lemmas, and provides a clear measure of the degree to which the approach can achieve the desired level of performance by relying solely on the overlap of forms. More importantly, the approach provides a clear failure mode, whereby errors can be attributed to either (i) lack of coverage (true OOV), or (ii) ambiguity (a given surface form is associated with multiple lemmas), rather than a lack of model transparency.

Let $\mathcal{D}_{\text{train}}$ denote the training set consisting of token–lemma pairs (w_i, ℓ_i). We induce a type-to-lemma mapping by selecting, for each word type w, the most frequent lemma observed with that type in the training data:

$$\hat{\ell}(w) \;=\; \arg\max_{\ell} \; \#\{(w_i, \ell_i) \in \mathcal{D}_{\text{train}} : w_i = w \wedge \ell_i = \ell\}. \tag{6.1}$$

This yields a dictionary M mapping each observed surface form to a single lemma: $M : w \mapsto \hat{\ell}(w)$.

The process is designed with a deliberately conservative approach, without character-level generalization or contextual information, thus focusing solely on the effect of lexical memorization.

Given a test token w_i, the baseline outputs:

$$\tilde{\ell}_i \;=\; \begin{cases} M(w_i), & \text{if } w_i \in \text{dom}(M) \\ w_i, & \text{otherwise (OOV pass-through).} \end{cases} \tag{6.2}$$

The OOV pass-through is utilized in the process to avoid making conjectural assumptions about the unknown forms. In historical datasets, there is a high amount of unknown forms that correspond to real orthographic novelty, editorial practices, or simply infrequent inflectional forms. The process is designed in such a manner that the errors remain understandable.

The dictionary baseline, as a measure, is naturally a significant comparator for Early Slavic, as it represents the most extreme form of the memorization-only approach, since the ability to look up and match the training and test forms yields a very precise result. However, the very features of Early Slavic that make it so fascinating as a language also highlight the limitations of the dictionary approach. As a result, the dictionary baseline is typically used as (i) a way to establish a bound on the level of generalization and (ii) a tool to measure the proportion of errors caused by coverage and ambiguity.

However, the process of dictionary lookup is only possible if the code points of the strings are the same. In other words, two strings will only be a match if their Unicode sequence is the same. This is a problem with the resources used in Early Slavic, as the same symbol may be represented differently due to the presence of private use area characters. Therefore, the dictionary baseline is always used after the standardization of the private use area characters.

There are benefits to the ordering: the process converts false out-of-vocabulary tokens into in-vocabulary types, stabilizes the frequency counts used in Equation 6.1, and ensures that the errors of the dictionary baseline capture more accurately linguistic variation.

Ambiguity represents a significant source of errors for Early Slavic, driven by syncretism and editorial normalizations, which either ignore or normalize the differences. While the most frequent lemma approach represents a reasonable default, it fails to incorporate contextual information and, as a result, inevitably selects the most frequent lemma, even if it is inappropriate for the context. This represents a key consideration for the forthcoming chapters, as it identifies a key objective for a model that supports contextual and character-level information.

To ensure the utility of the baseline, the dictionary lemmatizer is retained in its most basic form. However, it is desirable to distinguish the baseline, as implemented, from the optionally included and reported extensions. The implemented baseline strictly represents the most frequent lemma, including out-of-vocabulary pass-through. Optional extensions include support for part of speech, enabling a mapping from (word, UPOS) to lemma, as performed by hybrid approaches to the Universal Dependencies framework, e.g., Stanza; and the use of normalization-aware lookups, which can be constructed from a common standardization layer. Finally, the use of edit-pattern backoff rules, which can be constructed from pairs, represents a more nuanced extension and, as a result, shares a more significant degree of similarity to the edit rule lemmatizers.

Because the normalization component is shared by both the dictionary baseline and the neural model, the comparison between the two approaches is not confounded by encoding differences. Although the character-level network may automatically adjust to the encoding irregularities, it may overfit the encoding scheme without the use of the normalization component.

For the purpose of baseline evaluation, the conventional token-wise lemmatization methodology is utilized. In other words, the predicted lemmas and the gold lemmas are compared. This provides a measure of the correctness of the predicted lemmas, as measured by the annotation scheme. This methodology also enables the evaluation of the proposed model relative to the current hybrid and neural approaches. However, for the purpose of the current thesis, the focus of the evaluation methodology is the token-wise accuracy and robustness of the proposed model.

6.2 UDPipe 2

UDPipe is part of the well-known Universal Dependencies processing pipelines, which include tokenization, sentence segmentation, part-of-speech tagging, morphological tagging, dependency parsing, and lemmatization in one unified framework [89, 87]. Within the context of the Universal Dependencies framework, the relevance of the UDPipe processing pipeline to this thesis lies in its pedagogic value as a baseline due to its inherent design centered

upon the CoNLL-U format, along with its availability with pre-trained models for various treebanks, which facilitates evaluation in standardized conditions [90]. In this thesis, focus will be given to the lemmatization process offered by the UDPipe processing pipeline, with evaluation conducted with gold tokenization and gold sentence segmentation.

In the context of the UDPipe processing pipeline, the process of lemmatization has been formulated as a supervised transformation problem, whereby, given the input token and its candidate morphological analyses, the system predicts the lemma by applying the most likely lemma rule to the input token [88]. More specifically, candidate analyses may include a universal POS tag, a language-specific POS tag, morphological features, and a lemma rule, where the lemma rule maps the observed token to its corresponding lemma. These rules can be viewed as compact representations of character-level operations, such as the deletion of suffixes, replacement of endings, or alternation of stems. In this context, the UDPipe processing pipeline selects the most likely lemma rule for the given token with the help of a supervised model trained on the corresponding treebank. Two features are inherent to this processing pipeline that make it useful as a baseline for historical lemmatization.

Firstly, as the transformation is directly learned from the data, good performance is typically observed when the lemma formation is regular and the relevant edit patterns occur frequently enough in the training data. Secondly, as the prediction is based on the application of a rewrite rule, the prediction process is deterministic once the rule is fixed, which is useful for error analysis as the errors may be traced to the application of the wrong transformation class. However, the approach is limited by the diversity of edit patterns that the training data may contain. In historical datasets, where high surface-level diversity is common and many spelling variants occur, test forms may require transformation patterns that are absent or insufficiently represented in the training data, and the edit rule may not cover the range of morphological and spelling variations. Therefore, when assessing UDPipe's edit-rule-based lemmatization approach, it is useful to compare it with neural character-level lemmatization models, which may be more robust to unseen spelling and morphological variation.

6.3 Stanza

The Stanza NLP pipeline is developed by the Stanford NLP group, as described in [79], and is built upon the Stanford system submitted to the CoNLL-2018 shared task [78]. The Stanza pipeline is similar to the UDPipe pipeline in terms of functionality; it provides an end-to-end set of components for UD processing. However, the Stanza pipeline differs from the UDPipe in model architectures in that the Stanza lemmatization component is designed as a hybrid component with the explicit aim of using high-precision memorization of frequent forms and neural generalization of infrequent forms.

The Stanza lemmatization process is based on the integration of three components of evidence [78, 79]. First, the Stanza pipeline learns a context-sensitive lexicon based on the (FORM, UPOS) pair, where the most frequent lemma of the word is determined based on

the universal POS tag. This component of the Stanza pipeline is designed to cover the most common cases of ambiguity that correlate with POS tags, such as cases where the word is used as both a noun and a verb with different lemmas. Second, if the `(FORM, UPOS)` lookup fails, the Stanza pipeline uses a backoff lexicon based solely on the `FORM` of the word. This word-only lexicon is designed to cover cases where the UPOS-conditioned dictionary does not contain a matching entry. The backoff lexicon is a robust method of handling the most frequent forms in the training set. Third, the Stanza pipeline uses a Seq2Seq neural character-level model to generate the lemma string from the input word.

The neural part treats lemmatization as a character transduction problem and learns to map input strings to lemma strings. This is typically done using an encoder–decoder model with an attention mechanism [78, 79]. The former encodes the input token's character string to a representation, and the latter generates conditioned on the representation. This module allows Stanza to generalize to unseen inflected forms and, to some extent, to spelling variation, because the neural component relies on character-level regularities rather than only on memorized word forms.

The hybrid model has two advantages. For words with high frequency and stable spelling that are common in its training data, dictionary matching can achieve almost perfect accuracy without any generation errors. For words that are unseen in its training data, it can still generate a plausible lemma even if its surface form is not in its dictionary. At the same time, Stanza also inherits two weaknesses that are particularly relevant in Early Slavic languages. First, with small or very heterogeneous training data, the sparsity of the lexicon components may result in inconsistent lemma choices across resources, which diminishes the effectiveness of memorization. Second, with historical data showing significant spelling variations and uncommon characters, the generator must effectively generalize with little supervision, with performance decaying rapidly if orthographic changes result in input patterns deviating from the training distribution. These factors make Stanza an important and challenging comparison for OldSlavicLemma since both systems leverage a neural character-level transducer, with Stanza also leveraging dictionary mechanisms that can be beneficial (in the high-coverage regime) or detrimental (in the low-coverage regime).

In the comparison, we use gold tokenization and gold sentence segmentation with Stanza to ensure the evaluation conditions are comparable to those for OldSlavicLemma. For the raw-text-to-CoNLL-U comparison, Stanza is also used only for tokenization and sentence segmentation before applying OldSlavicLemma to the produced CoNLL-U files. Therefore, any differences can be attributed to the lemmatization process. For UD v2.12, we include the official results for Stanza if available; for UD v2.15, we include Stanza results with the corresponding models using gold tokenization evaluation like the results for our proposed system. In addition, we include the Stanza model versions and the corresponding UD release identifiers to guarantee reproducibility of the comparison.

6.4 Shared-Task Systems (SIGTYP 2024)

To contextualize our results outside the realm of Early Slavic and to compare with competitive recent approaches to historical lemmatization, we include baselines from the SIGTYP 2024 shared task [28] on word embedding evaluation for ancient and historical languages, which included lemmatization among its evaluation tasks. The shared task is particularly relevant to this thesis, as it includes many settings that this research is designed to address: limited training data, significant variability in spelling conventions across languages and writing systems, and assessment based on exact string matching rather than secondary metrics. Moreover, the shared task paradigm provides a unified testing framework, which includes data splits, scoring scripts, and reporting conventions. This reduces the risk of comparability concerns that often arise from the use of varying preprocessing or scoring strategies across individual papers.

Our performance assessment of OldSlavicLemma is conducted with respect to the top-performing approaches reported in the shared task results [28], such as HDB–BOS [81], Team 21a [64], UDParse [41], and TartuNLP [30]. Though they vary significantly in their underlying architecture, the approaches identified above illustrate three recurring architectural patterns in the SIGTYP 2024 systems: (i) multilingual Transformer-based solutions, (ii) parameter-efficient fine-tuning strategies designed to reduce overfitting with limited training data, and (iii) character-aware or hybrid architectures that address morphological transformation.

Several SIGTYP 2024 systems relied on pretrained Transformer-based models or Transformer-style encoders, which were then adapted or fine-tuned for the shared-task settings [28]. The underlying rationale for this approach is that pre-training allows for the integration of lexical and contextual information that might not be present within the training set, and sub-word tokenization can mitigate the problem to a certain extent. However, for historical languages, pre-training might be problematic since differences in script, spelling, and content might reduce its effectiveness. Additionally, sub-word tokenization might fragment historical spelling and lead to idiosyncratic sub-word forms that might reduce the effectiveness of the model. For this reason, some SIGTYP 2024 systems incorporated character-aware modeling, for example, through hierarchical tokenization or character-level generation for lemmatization [81].

Another important pattern in the competition is parameter-efficient fine-tuning (PEFT), which involves the fine-tuning of a limited set of parameters within a pre-trained model [30, 41]. The rationale for this approach is that the training set for historical languages might be too limited to allow for the updating of a large number of parameters without risking overfitting. This strategy is useful because it balances limited task-specific supervision with the large capacity of a pre-trained model.

The third pattern focuses on character-aware modeling and the explicit treatment of morphological transduction, sometimes in combination with contextual encoders. Methods in this group approach lemma prediction as a structured string transformation, aiming to capture regularities in string edits at the character level, which is often the appropriate inductive bias for morphologically rich languages and historical spelling. In the SIGTYP 2024 constrained

setting, character-aware modeling was particularly relevant, as illustrated by the HDB–BOS system, which used character-aware hierarchical Transformers and character-level T5 models for lemmatization [81].

In this thesis, the results provided by the SIGTYP 2024 shared task are used as a reference point for the performance of shared-task systems, as the shared-task organizers provide a fixed evaluation protocol and scores for the participating teams. It is worth noting that SIGTYP 2024 distinguishes between constrained and unconstrained settings, where the unconstrained setting allows the use of external resources and pre-trained models. The primary comparison for OldSlavicLemma is made in the constrained setting, because our model is trained only on the provided task data. However, selected unconstrained systems are also reported as additional reference points, while clearly marking them as such.

For scoring, we follow the official SIGTYP evaluation process and report scores generated by the shared task scoring program [28]. To be more precise, SIGTYP reports Accuracy@1 and Accuracy@3 scores, which are then combined following the official evaluation protocol (Section 7.2). Using the official evaluation script helps to avoid subtle normalization and string comparison issues that may occur due to the historical corpora's use of diacritics and non-standard character combinations. In reporting results against the shared task systems, we (i) quote the shared task scores for the systems we are interested in, and (ii) compute the scores using the official evaluation script, which we use for scoring in this work, to ensure that the results we observe are due to the models rather than evaluation differences.

In summary, the SIGTYP baselines play two roles in our evaluation process. Firstly, they allow us to compare OldSlavicLemma with historical lemmatization systems outside the Slavic family, including systems not specifically designed for Early Slavic. This helps determine whether the observed gains are specific to Early Slavic or reflect broader benefits of the proposed architecture. Secondly, the baselines serve as a sanity check with respect to typological and orthographic variation, given the diverse range of languages included in the benchmark [28]. This gives us a much larger context than evaluation across Early Slavic UD treebanks only, and supports the claim that historical lemmatization models should be evaluated not only in terms of in-domain performance but also with respect to their ability to generalize to new, unseen data.

6.5 Retrained Settings for the Newly Annotated OCS Benchmark

In addition to the off-the-shelf Stanza model, two retrained Stanza and OldSlavicLemma settings were used for benchmarking on the newly annotated OCS material. The purpose of these settings was to test whether adaptation to the new annotation improves lemmatization performance and to assess the effect of combining the new data with the existing UD OCS resource.

To support standard evaluation, the annotated OCS material was partitioned into training,

development, and test sets using an approximate 80/10/10 split at the sentence level, while preserving sentence boundaries within each source file.

The train/dev/test split of the newly annotated OCS benchmark follows the protocol described in Chapter 7: 23,768 training tokens, 2,979 development tokens, and 2,931 test tokens.

The first retrained setting uses only the newly annotated OCS material introduced in this thesis. The corpus was divided into training, development, and test portions, and models were retrained on the training split with model selection based on the development split. This setting is intended to measure in-domain adaptation to the orthographic, lexical, and annotation properties of the new resource.

The second retrained setting uses a combined training corpus formed by merging the new annotated OCS training data with the existing UD v2.12 OCS training data. This setting was included to test whether broader lexical coverage and exposure to previously available OCS supervision improve generalization, both on the new benchmark and on the standard UD OCS test material.

In both retrained settings, we use the Stanza and OldSlavicLemma lemmatizer within the same general pipeline framework as the pretrained system, while changing only the training data configuration. This makes the comparison more interpretable: differences in performance can be attributed primarily to the effect of retraining data.

Chapter 7

Experimental Protocol

This chapter describes the experimental protocol used for evaluating lemmatization for Early Slavic texts in a manner that is (i) directly comparable to existing UD pipelines and benchmarks, and (ii) informative regarding the error patterns that are characteristic of historical corpora. While, in modern standardized language environments, corpora are characterized by full supervision, little variability in spelling, and a relatively small set of word forms, in historical corpora, we are confronted with a situation in which a single score does not capture the differences between systems, such as whether improvements are driven by memorization of frequent word forms, actual generalization to unseen word forms, or disambiguation of contextual information. The protocol therefore combines standard benchmark-style evaluation with targeted analyses designed to expose robustness properties that are especially relevant for OCS and OES.

We structure the evaluation around four complementary settings. First, we report *in-family* evaluation on Early Slavic treebanks (OCS and OES), which are the main domain of interest in this thesis. Second, we evaluate *cross-family historical generalization* on the SIGTYP 2024 benchmark [28]. Third, we report *broad multilingual generalization* on a curated subset of UD v2.12 treebanks. Fourth, we include a newly annotated OCS benchmark introduced in this thesis, which is used to evaluate transfer to newly curated material and the effect of retraining on in-domain annotation.

The ReduceLROnPlateau scheduler is used to lower the learning rate when validation accuracy ceases to improve, and early stopping is applied accordingly. The best checkpoint is selected based on accuracy on the validation set. All experiments were run on an NVIDIA GeForce RTX 4060 GPU.

In the main settings, we evaluate under gold tokenization and gold sentence segmentation as supplied by the benchmark resources. For the Early Slavic UD v2.12 and v2.15 treebanks, we additionally report a raw-text-to-CoNLL-U setting. In this setting, raw text is tokenized with Stanza, and the resulting CoNLL-U files are used as input for OldSlavicLemma. This approach to evaluation was intentionally chosen to separate lemmatization from other processes such as segmentation, as well as to avoid measuring the quality of lemmatization in conjunction with tokenization errors. This approach to evaluation is similar to the dominant

reporting convention used in benchmarking the universal dependencies (UD) pipeline, thus facilitating comparison with publicly reported results for UDPipe and Stanza, when such results are publicly available [90, 79], unless otherwise specified.

7.1 Datasets and Splits

The section is organized as follows. First, we describe UD as the common evaluation substrate used to represent the main treebank data and to support comparison with existing systems. We then introduce the Early Slavic treebanks used in the in-family experiments, followed by the SIGTYP 2024 historical benchmark, the multilingual UD subset, and the newly annotated OCS benchmark. For each setting, we specify the role of the dataset in the evaluation and the way in which the corresponding splits are used.

7.1.1 Universal Dependencies as a Common Evaluation Substrate

A significant portion of the experiments carried out in this thesis relies on the UD Initiative, an open, cross-linguistic project that provides a unifying annotation scheme for morphological and syntactic structure, as well as language-specific information when necessary [69, 27]. UD treebanks are distributed in the CoNLL-U format, which provides token-level columns for surface form (`FORM`), lemma (`LEMMA`), universal and language-specific POS tags (`UPOS`, `XPOS`), morphological features (`FEATS`), and dependency information (`HEAD`, `DEPREL`) [69]. In the context of this thesis, this representation serves three methodological roles.

First, it provides a uniform interface for training and evaluation across both Early Slavic and multilingual data. Because the lemma is defined at the token level and aligned to gold tokenization, lemmatization can be evaluated consistently across treebanks without dataset-specific evaluation code. Second, the UD format supports *fair comparison* with established UD pipelines such as UDPipe and Stanza, which are designed around CoNLL-U and are commonly reported under the same evaluation assumptions [90, 79]. Third, CoNLL-U retains contextual signals—POS and morphological features—that are useful for analysis. Even when the proposed model is not conditioned on these fields, they enable controlled breakdowns (e.g., by POS class) and facilitate interpretation of errors typical of historical corpora.

7.1.2 Early Slavic Treebanks (UD v2.12 and UD v2.15)

The focus of the in-family evaluation is on Early Slavic treebanks that include lemma annotation for OCS (`chu`) and OES (`orv`). OCS is predominantly represented by the OCS-PROIEL treebank in UD v2.12, and for OES, we use the treebank family that is based on the treebank part of the Tromsø Old Russian and OCS Treebank (TOROT) corpus, as well as all extensions thereof, including Russian National Corpus (RNC) and Birchbark. These treebanks cover various genres and editorial traditions, reflecting the main difficulties in historical lemmatization: complex inflectional morphology, syncretism, and significant surface form variation.

To assess robustness to resource updates and annotation drift, we report results on corresponding Early Slavic treebanks in UD v2.15 as well, which contains updates that include corrections to treebank content, such as tokenization and lemma updates, as well as new sources added for OES treebanks relative to v2.12.[1] We always use the *canonical train/dev/test splits* that come with treebanks, as well as preserving gold tokenization or segmentation.This choice ensures that performance differences are attributable to lemmatization behavior rather than to altered preprocessing.

7.1.3 SIGTYP 2024 Historical Benchmark

To evaluate cross-family historical generalization and to compare against shared-task systems, we use the SIGTYP 2024 benchmark for ancient and historical language processing [28]. The SIGTYP 2024 dataset contains 13 languages, mostly from the UD v2.12, with a standardized evaluation framework for the task of lemmatization across various script types and families of languages. The SIGTYP 2024, following the shared task framework, utilizes the 80/10/10 split (train/validation/test) at the sentence level for the training, validation, and test sets, respectively.

The SIGTYP 2024 specifies two training settings: the **constrained** track (C), in which the system is restricted to use only training data, and the **unconstrained** track (U), in which external data resources and pretrained models are permitted [28]

In this thesis, the results of the experiments are presented in the **constrained** regime. This is in line with the underlying motivation for historical NLP and the focus of this thesis on robust lemmatization with limited supervision and diverse data sources. The evaluation metrics are calculated with the *official* SIGTYP scoring program,[2], thus ensuring strict comparability with the official results of the shared task and reducing potential divergences in string matching and normalization, especially in diacritic-rich historical data.

7.1.4 Multilingual UD v2.12 Benchmark Subset

Beyond historical assessment, we examine the model's ability to generalize across a broad multilingual setting by evaluating on a curated subset of the UD v2.12 treebanks. The selection of these treebanks is motivated by the widespread adoption of the data in published evaluations of mainstream UD pipelines, with official results for lemmatization of systems like UDPipe 2 and Stanza routinely reported for this release of the data [90, 79]. Using the same release of the data helps ensure that any improvements are not due to changes in the data itself between releases.

Because the UD includes treebanks with differences in size, genre, and annotation history, an indiscriminate "all treebanks" evaluation may be misleading, as small treebanks may cause unstable training behavior for Seq2Seq models, and some treebanks may not have comparable

[1] Note that when we report results on specific releases, we always use the corresponding version of UD treebanks to ensure that we do not leak information across annotation versions.

[2] `https://github.com/sigtyp/ST2024/blob/main/scoring_program_constrained.zip`

baselines reported. Therefore, the multilingual subset is limited to treebanks fulfilling the following comparability criteria:

1. canonical train/dev/test splits are defined in UD v2.12;
2. lemmatization scores for at least one major UD pipeline baseline (Stanza and/or UDPipe 2) are publicly reported under comparable evaluation assumptions;
3. the training set is sufficiently large to train character-level Seq2Seq models without extreme instability (we exclude extremely small treebanks).

This filtered benchmark spans typologically diverse languages and includes both morphologically simpler and richer systems. It provides a broad stress test for the proposed model's generalization and supports later interpretation of relative strengths and weaknesses in relation to training size, morphological complexity, and orthographic stability.

7.1.5 Newly Annotated OCS Benchmark

In addition to standard benchmark datasets, this thesis evaluates lemmatization on a newly annotated OCS dataset constructed as part of the resource-development workflow described in earlier chapters. This dataset is included as a dedicated robustness and adaptation benchmark, with the goal of testing how well existing systems transfer to newly curated material that differs from standard UD resources in source composition, orthographic profile, and annotation history.

To support controlled evaluation, the annotated OCS material was partitioned into training, development, and test sets using an approximate 80/10/10 split at the sentence level, while preserving sentence boundaries within each source file, and sentence integrity was preserved when available. Some dictionary-style entries were converted into CoNLL-U as single-token sentences, and therefore, the resulting token counts are approximate. In the current version of the dataset, this corresponds to 23,768 training tokens, 2,979 development tokens, and 2,931 test tokens. Moreover, the retrained lemmatizers rely on silver POS tags from the official Stanza POS tagger.

Evaluation on this benchmark is performed under gold tokenization using exact-match lemma accuracy. In addition to off-the-shelf systems, we also evaluate retrained settings in order to distinguish zero-shot transfer from in-domain adaptation. In particular, we compare (i) pretrained systems applied directly to the new OCS benchmark, (ii) systems retrained only on the newly annotated OCS training split, and (iii) systems retrained on a combined training set formed by merging the new OCS training data with the UD v2.12 OCS training data. This design makes it possible to assess both benchmark difficulty and the value of newly annotated in-domain supervision.

7.2 Evaluation Metrics and Scoring Procedures

This section defines the evaluation metrics and scoring procedures used to compare the lemmatization systems across the experimental settings. Since the experiments include both gold-tokenized and raw-text-to-CoNLL-U evaluation, the scoring procedure must distinguish between errors caused by lemma prediction and errors introduced by tokenization, sentence segmentation, or format conversion. For gold-tokenized evaluation, the main metric is lemma-level accuracy. For raw-text-to-CoNLL-U evaluation, we report the official Lemmas F1 score produced by the CoNLL shared-task evaluation script, since token boundaries and alignment may differ. For the SIGTYP 2024 experiments, we follow the official shared-task scoring protocol based on Accuracy@1 and Accuracy@3. In addition to these aggregate metrics, the evaluation also considers specific token subsets, including OOV and ambiguous forms, in order to assess robustness beyond overall accuracy.

7.2.1 Lemmatization Evaluation Metrics

For experiments under gold tokenization and gold sentence segmentation, we report lemma accuracy as the primary metric. A prediction is counted as correct if and only if the predicted lemma string exactly matches the gold lemma string for the aligned token. Given a test set of N tokens with gold lemmas ℓ_i and predicted lemmas $\hat{\ell}_i$, EM is defined as

$$\text{EM} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\{\hat{\ell}_i = \ell_i\}. \tag{7.1}$$

We adopt accuracy because, under gold tokenization and gold sentence segmentation, token boundaries are fixed and the task reduces to token-level lemma prediction. We verified with the official CoNLL 2018 UD evaluation script that, in this token-aligned setting, lemma precision, recall, F1, and aligned lemma accuracy are numerically the same. Therefore accuracy provides the clearest and most transparent way to report lemmatization quality under gold segmentation.

By contrast, for raw→CoNLL-U end-to-end evaluation, tokenization and sentence segmentation are not fixed. In our case, this setting uses the Stanza tokenizer, while lemma prediction is performed by OldSlavicLemma. In that setting, we report the official Lemmas F1 score produced by the CoNLL shared-task evaluation script.

7.2.2 SIGTYP Scoring: Accuracy@1 and Accuracy@3

For the SIGTYP 2024 task, we follow the official evaluation protocol [28]. The scorer calculates the Accuracy@1 and Accuracy@3 metrics, returning their unweighted mean as the final score. Accuracy@1 requires an exact match with the top prediction, while Accuracy@3 credits a system if the lemma appears in the top three predictions. We report the final score for the SIGTYP 2024 task as computed by the official scorer, as this ensures comparability

with shared task participants and minimizes the risk of metric implementation differences.

7.2.3 Token-Category Analyses: Seen/OOV and Ambiguity

Aggregate accuracy may not reveal qualitatively different sources of errors. This problem is especially prevalent in historical data, as the most common failure type may depend on the degree of orthographic variation and the level of supervision. In order to make these effects explicit, we provide category-based analyses on the UD test data with two different partitions of the training data:

1. **Seen vs. OOV:** a token is *seen* if its surface form occurs at least once in training; otherwise it is OOV. This split quantifies how much performance depends on lexical overlap and how well the model generalizes to unseen forms.

2. **Ambiguous vs. unambiguous types:** a surface form type is *ambiguous* if it is aligned to multiple lemma values in training; otherwise it is *unambiguous*. This split isolates the effect of lemma ambiguity and probes whether context-sensitive modeling improves disambiguation.

We compute accuracy on each subset, and the results are particularly insightful for Early Slavic because (i) orthography increases OOV rates beyond those found in standard modern datasets, and (ii) syncretic morphology and homography create actual ambiguities even within seen types. These subsets provide a more accurate picture of historical lemmatization quality than overall EM analysis.

Chapter 8

Results and Analysis

This chapter describes and analyzes the performance of OldSlavicLemma on Early Slavic UD treebanks, multilingual UD data, and historical shared task data. We also discuss robustness to spelling variation and domain adaptation, study the contribution of individual components using ablation studies, and conduct an error analysis to determine the cause of any remaining errors.

8.1 Main Results on Early Slavic Treebanks

First, we consider the case of Early Slavic languages, with a focus on OCS and OES. These languages are characterized by complex inflectional morphology, significant orthographic variation, and a lack of supervised data, which makes lemmatization particularly sensitive to sparsity and out-of-vocabulary words. Table 8.1 reports Lemma Accuracy for gold-tokenized experiments and official Lemmas F1 for raw→CoNLL-U end-to-end evaluation.

Table 8.1: Lemmatization performance (%) on historical Slavic UD v2.12 test sets. Gold denotes lemma accuracy with gold tokenization, while raw denotes Lemmas F1 using the end-to-end CoNLL-2018 evaluation script under automatic tokenization.

Language	**Treebank**	**OCS-DBA**	**Gold Acc.**			**Raw F1**		
		Gold Acc.	**Stanza**	**UDPipe 2**	**Our model**	**Stanza**	**UDPipe 2**	**Our model**
Old Church Slavonic	PROIEL	78.19	94.75	90.19	**96.38**	94.65	90.18	**96.30**
Old East Slavic	Birchbark	55.05	73.89	65.59	**76.60**	73.69	65.58	**76.47**
Old East Slavic	RNC	64.65	81.64	78.28	**83.04**	81.04	77.22	**82.58**
Old East Slavic	TOROT	77.87	92.82	88.14	**94.52**	92.79	88.09	**94.47**

The raw-text results are close to the gold-tokenized scores in most cases, showing that the proposed lemmatizer remains stable when used with existing tokenizers.

The results for UD v2.12 show that OldSlavicLemma achieves the highest lemma score on all Early Slavic treebanks. The largest gains occur under conditions with significant orthographic noise and unseen spellings, especially on OES–Birchbark. This supports our intuition that our model is helped by transducing characters given contextual information rather than relying on stored knowledge. Our model is compared to hybrid models like Stanza [79],

which uses both dictionary and neural approaches. Our model outperforms Stanza by being dictionary-free and achieving good results on seen forms and better generalization on unseen forms.

Table 8.2: Lemmatization performance (%) on Early Slavic UD v2.15 treebanks. Gold denotes lemma accuracy with gold tokenization, while raw denotes Lemmas F1 using the end-to-end CoNLL-2018 evaluation script under automatic tokenization.

Language	**Treebank**	**OCS-DBA**	**Gold Acc.**			**Raw F1**		
		Gold Acc.	**Stanza**	**UDPipe 2**	**Our model**	**Stanza**	**UDPipe 2**	**Our model**
Old Church Slavonic	PROIEL	78.19	94.35	90.29	**96.28**	94.32	90.21	**96.18**
Old East Slavic	TOROT	77.87	92.33	88.48	**94.48**	92.31	88.42	**94.43**
Old East Slavic	Birchbark	55.05	74.04	65.28	**75.93**	73.83	65.21	**75.88**
Old East Slavic	RNC	79.78	92.72	90.89	**94.09**	92.58	90.63	**93.86**
Old East Slavic	Ruthenian	72.41	89.51	82.96	**91.05**	89.46	82.89	**91.01**

In order to verify the stability of the improvement under changing annotations and resource revisions, we also conduct evaluation on UD v2.15, as reported in Table 8.2. The performance is consistent: OldSlavicLemma outperforms the previous state-of-the-art tools UDPipe 2 [90], Stanza [79], and the dictionary-based approach OCS-DBA [68] over all the available Early Slavic treebanks. This confirms that the method is not overfitted to a specific UD release and remains stable under changes to lemma inventories and token–lemma alignments.

8.2 Multilingual Generalization Across UD Treebanks

Early Slavic is the primary focus of this thesis, but we also evaluate generalization beyond Slavic to understand whether the architecture transfers to typologically diverse languages and scripts. We report results on (i) a historical multilingual benchmark (SIGTYP 2024), and (ii) a large comparability-oriented subset of UD v2.12 treebanks where official UDPipe 2 and Stanza results are publicly available.

8.2.1 Historical Multilingual Benchmark (SIGTYP 2024)

The SIGTYP 2024 shared task introduced a standardized evaluation suite for ancient and historical languages, derived primarily from UD v2.12, with an 80/10/10 split by sentences and two tracks: constrained (training only on provided data) and unconstrained (allowing external resources/pretraining) [28]. We train OldSlavicLemma under the constrained protocol and compare against top shared-task systems including HDB–BOS [81], Team 21a [64], UDParse [41], and TartuNLP [30].

Overall, OldSlavicLemma outperforms the strongest reported baselines on the majority of languages in this benchmark, despite using only constrained training data, as reported in Table 8.3. The largest gains appear in morphologically rich historical languages with non-trivial orthographic variation (including Early Slavic, Gothic, and Sanskrit). In contrast, lan-

Table 8.3: Lemmatization score by language and system on the SIGTYP 2024 dataset. Scores for the four systems and baseline are taken from Table 5 of SIGTYP 2024 findings [28], and bold indicates the best result in each row. Arrows in the last column indicate whether our model performs better (↑) or worse (↓) than the best baseline. Δ shows the absolute difference between our model and the best baseline, while (C) and (U) indicate constrained and unconstrained protocols, respectively.

Language	Baseline	HDB–BOS (C)	Team 21a (C)	UDParse (U)	TartuNLP (U)	Our Model (C)	Δ vs. Best
♣ Ancient Greek	91.06	94.08	88.30	94.02	93.39	**97.81**	+3.73 ↑
♢ Ancient Hebrew	95.28	97.29	61.75	96.85	98.15	**99.39**	+1.24 ↑
♠ Classical Chinese	98.81	99.18	99.84	**99.96**	99.91	97.66	–2.30 ↓
♢ Coptic	95.74	95.07	46.32	74.78	**98.28**	97.80	–0.48 ↓
♣ Gothic	91.95	93.31	90.79	92.81	95.41	**97.95**	+2.54 ↑
♣ Medieval Icelandic	93.78	96.63	94.58	97.96	97.23	**98.03**	+0.07 ↑
♣ Classical & Late Latin	92.08	96.00	92.35	96.74	96.99	**97.94**	+0.95 ↑
♣ Medieval Latin	97.03	98.46	97.22	98.91	98.69	**99.07**	+0.16 ↑
♣ Old Church Slavonic	89.60	94.49	79.59	59.56	92.70	**98.03**	+3.54 ↑
♣ Old East Slavic	84.44	90.09	78.44	68.55	89.23	**95.22**	+5.13 ↑
♣ Old French	91.93	92.63	83.32	92.47	95.11	**97.29**	+2.18 ↑
♣ Vedic Sanskrit	84.24	84.59	83.21	88.10	91.48	**95.68**	+4.20 ↑
♡ Old Hungarian	89.43	85.92	69.97	63.43	86.91	**97.28**	+7.85 ↑

guages where unconstrained transformer-based systems reach near-ceiling performance (e.g., Classical Chinese) remain challenging for a purely task-trained recurrent model, suggesting that pretraining and segmentation conventions can dominate performance when labels are abundant or the evaluation is saturated.

8.2.2 Broad UD v2.12 Multilingual Evaluation

We further evaluate on a large subset of UD v2.12 treebanks to test transfer beyond historical datasets. To ensure comparability, we include only treebanks that (i) provide canonical train/dev/test splits, and (ii) have publicly reported UDPipe 2 and Stanza lemmatization scores for UD v2.12 under the same evaluation conditions. This yields a broad multilingual testbed spanning many families and scripts. Table 8.4 reports the broad UD v2.12 multilingual evaluation.

Across this benchmark, performance is generally competitive, and the model's areas of relative strength lie within historically variable and morphologically rich contexts, especially those of Indo-European historical corpora. Degradation is seen in treebanks where (a) lemmatization is closely tied to pipeline-specific tokenization or segmentation choices, or (b) morphology is strongly non-concatenative or templatic, allowing edit-rule systems and transformer pipelines to take advantage of additional information or segmentation priors. Notably, the Early Slavic results are seen as among the strongest areas of improvement, aligning with the thesis focus.

8.3 Robustness to Orthographic Variation and Domain Shift

A central motivation for this thesis is that historical corpora contain systematic and unsystematic variation: spelling alternations, scribal conventions, diacritic usage, abbreviation marks, and editorial normalization differences. Domain variation compounds this: formal chronicle-

Table 8.4: Dictionary denotes lemma accuracy with gold tokenization, while raw denotes F1 score using end-to-end evaluation with automatic tokenization.

Language	Treebank	Total words	Dictionary	Stanza RAW	UDPipe 2 RAW	Our RAW
Afrikaans	AfriBooms	49260	94.48	**98.29**	**98.29**	97.96
Ancient Greek	PROIEL	214005	89.20	97.18	94.76	**98.17**
Ancient Greek	Perseus	202989	74.84	87.86	86.72	**90.04**
Ancient Hebrew	PTNK	39036	87.66	**92.14**	67.47	91.63
Arabic	PADT	282384	91.95	93.54	90.34	**93.64**
Armenian	BSUT	41805	75.86	94.99	**96.79**	94.09
Armenian	ArmTDP	52585	78.48	91.35	**94.93**	94.90
Basque	bdt	121443	84.18	96.38	96.38	**96.61**
Belarusian	HSE	305109	81.97	94.59	93.25	**94.61**
Bulgarian	BTB	156149	88.53	97.28	**98.01**	97.54
Catalan	AnCora	546665	96.94	99.06	**99.40**	99.19
Chinese (Simplified)	GSDSimp	123291	99.90	92.07	90.23	**92.49**
Chinese (Traditional)	gsd	123291	99.90	**92.70**	90.20	92.28
Classical Chinese	Kyoto	433168	99.74	**97.63**	97.51	95.06
Coptic	Scriptorium	55858	90.23	85.95	73.90	**86.14**
Croatian	SET	199409	87.95	96.60	**97.78**	96.82
Czech	FicTree	167056	89.28	98.32	**99.29**	98.37
Czech	CAC	494383	89.29	97.46	**99.30**	97.59
Czech	CLTT	36013	84.44	95.78	**98.48**	95.62
Czech	PDT	1506486	94.05	98.73	**99.38**	98.71
Danish	DDT	100733	90.05	**97.88**	97.38	96.09
Dutch	Alpino	208614	90.77	**95.33**	94.91	94.70
Dutch	Lassysmall	98107	91.64	95.86	**96.18**	94.96
English	Atis	61879	99.39	**99.65**	99.63	99.57
English	EWT	254819	95.15	96.63	**97.26**	95.69
English	GUM	187416	95.92	98.22	**98.84**	97.56
English	LinES	94217	94.79	98.26	**98.33**	97.80
English	ParTUT	49633	94.40	97.31	**98.17**	97.08
French	GSD	400289	95.92	**98.20**	97.72	97.90
French	ParTUT	28576	92.93	97.35	**97.89**	96.58
French	Rhapsodie	44242	92.90	97.74	**98.19**	96.37
French	Sequoia	70546	94.55	**98.50**	98.30	97.97
French	ParisStories	42795	94.55	97.66	**98.02**	95.94
Estonian	EDT	437801	84.63	95.93	95.45	**96.17**
Estonian	EWT	90586	80.42	90.77	**93.75**	91.64
Finnish	FTB	159612	79.25	**97.10**	95.76	96.39
Finnish	TDT	202192	79.12	**95.98**	92.11	94.98
Gothic	PROIEL	55336	88.17	96.10	94.71	**96.88**
Greek	GDT	63441	86.98	96.05	96.09	**96.51**
Galician	CTG	138837	92.19	98.01	**98.12**	97.78
German	GSD	292769	94.05	**97.23**	96.91	96.56
German	HDT	3455580	95.46	**98.04**	97.69	95.86
Hindi	HDTB	351704	95.71	96.88	**98.93**	98.70
Hebrew	IAHLTwiki	140950	92.39	**92.51**	87.15	91.49
Hebrew	HTB	160195	91.18	**89.71**	83.02	89.14
Hungarian	Szeged	42032	77.70	93.81	**94.89**	93.90
Irish	twittirish	47790	85.53	87.96	**88.41**	87.35
Irish	IDT	115990	90.21	95.32	**95.85**	95.55
Icelandic	GC	99611	78.35	91.06	**91.64**	89.17
Icelandic	IcePaHC	985050	90.44	95.78	**96.24**	95.39
Icelandic	Modern	80395	88.32	95.34	**96.87**	94.95
Indonesian	GSD	122019	89.53	**98.28**	98.12	97.96
Italian	postwita	124515	93.06	**96.63**	96.62	95.96
Italian	ParTUT	55558	92.31	97.71	**98.57**	97.29
Italian	MarkIT	40488	81.71	87.08	**88.18**	86.01
Italian	ISDT	298337	94.93	98.23	**98.68**	97.99
Italian	VIT	280145	94.45	98.35	**98.87**	97.76
Italian	twittiro	29602	88.14	92.91	**94.57**	91.50
Japanese	GSD	193654	95.76	**96.02**	95.03	95.03
Japanese	GSDLUW	150243	94.94	**94.67**	93.56	91.96

Language	Treebank	Total words	Dictionary	Stanza RAW	UDPipe 2 RAW	Our RAW
Korean	Kaist	350090	72.73	94.10	**94.18**	93.45
Korean	GSD	80322	71.53	93.55	**93.62**	93.43
Latin	PROIEL	205566	88.67	**96.86**	96.19	96.82
Latin	LLCT	242411	96.06	98.09	97.76	**98.52**
Latin	UDante	55524	71.86	86.95	**86.97**	86.83
Latin	ITTB	450515	96.13	99.09	**99.16**	98.54
Latvian	LVTB	289062	85.59	96.12	**96.57**	96.26
Lithuanian	ALKSNIS	70047	70.79	92.95	**93.45**	92.71
Maghrebi Arabic	Arabizi	19793	50.07	51.03	50.63	**52.05**
Marathi	UFAL	3847	81.80	82.84	**84.91**	76.72
Naija	NSC	140818	98.68	99.14	**99.33**	99.26
Norwegian	Bokmaal	310221	94.06	**98.61**	98.58	97.91
Norwegian	Nynorsk	301353	93.34	97.87	**98.46**	97.50
Old Church Slavonic	PROIEL	198843	78.19	94.65	90.18	**96.30**
Old East Slavic	Birchbark	27269	55.05	73.69	65.58	**76.47**
Old East Slavic	RNC	48647	64.65	81.04	77.22	**82.58**
Old East Slavic	TOROT	246850	77.87	92.79	88.09	**94.47**
Persian	PerDT	501776	97.77	98.55	**98.91**	98.46
Persian	Seraji	152920	96.83	98.18	**98.26**	98.00
Portuguese	Bosque	227827	92.78	98.04	**98.27**	97.81
Portuguese	CINTIL	475860	94.14	**98.56**	97.66	97.96
Portuguese	PetroGold	250605	96.02	99.02	**99.09**	98.81
Pomak	Philotis	86780	91.58	97.90	96.71	**98.28**
Polish	LFG	130967	84.33	97.24	**98.24**	96.76
Polish	PDB	350036	88.20	97.47	**98.08**	97.25
Romanian	Nonstandard	572436	89.61	**94.96**	94.82	94.93
Romanian	RRT	218522	89.94	97.42	**98.00**	97.38
Romanian	simonero	146020	92.80	98.69	**98.89**	98.68
Russian	GSD	97994	79.72	95.51	**96.87**	94.46
Russian	Taiga	197001	82.82	94.41	**94.62**	93.99
Russian	SynTagRus	1515683	91.73	97.65	**98.19**	97.46
Serbian	SET	97673	86.91	96.30	**97.82**	96.79
Scottish Gaelic	ARCOSG	89958	92.81	95.81	94.92	**96.19**
Spanish	AnCora	559782	96.72	99.24	**99.39**	99.20
Spanish	GSD	431584	93.29	98.42	**98.58**	98.20
Swedish	LinES	90960	90.29	97.19	**97.82**	96.23
Swedish	Talbanken	96820	89.32	97.92	**98.17**	97.20
Slovenian	SSJ	267097	87.17	97.69	**98.57**	97.48
Slovak	SNK	106097	69.31	94.08	**96.51**	93.97
Tamil	TTB	9581	72.95	82.55	**88.79**	81.47
Turkish German	SAGT	37227	84.48	90.24	**90.80**	90.42
Turkish	Atis	45875	97.44	98.98	**99.11**	98.52
Turkish	BOUN	125212	77.84	90.37	90.47	**91.02**
Turkish	FrameNet	19221	69.67	95.16	**96.39**	93.39
Turkish	IMST	58096	77.55	**95.55**	94.46	94.96
Turkish	Penn	183555	86.83	**94.58**	94.14	94.21
Turkish	kenet	178658	80.58	93.22	**93.50**	93.31
Turkish	Tourism	91469	96.32	98.90	98.27	**99.02**
Urdu	UDTB	138077	94.78	95.58	**97.41**	97.16
Ukrainian	IU	122750	80.18	96.72	**97.47**	96.18
Uyghur	UDT	40236	64.13	95.30	94.71	**95.31**
Vietnamese	VTB	58069	97.78	81.66	85.76	**87.55**
Western Armenian	ArmTDP	122907	84.99	96.70	**97.14**	96.61
Welsh	CCG	48530	90.34	94.85	94.43	**95.01**
Wolof	wtb	44258	91.96	94.63	**95.18**	94.85

like texts differ substantially from vernacular documents and short fragments (e.g., birch-bark letters). The OES family of treebanks provides a natural stress test: TOROT contains more formal genres; RNC broadens domains; and BIRCHBARK is highly informal and fragmentary.

The pattern observed in the results for Early Slavic indicates that OLDSLAVICLEMMA has certain robustness properties in cases where there is significant orthographic drift and domain shift, resulting in many unseen surface forms. The fact that the model is at the character level and that the lemma generation is context-dependent means that the model generalizes by stem and suffix alternations, as well as context-dependent disambiguation, rather than direct coverage of the lexicon. This is consistent with the strongest relative gains appearing in BIRCHBARK and other high-variation subsets.

8.3.1 Additional Evidence from Newly Annotated OCS Material

To complement the main Early Slavic experiments, we also report results on a newly annotated OCS test set constructed during this thesis. Since the corpus construction, normalization, and annotation workflow have already been described in previous chapters, the purpose of this subsection is simply to show the empirical value of that effort for lemmatization evaluation. In particular, these results test whether existing OCS lemmatizers transfer to newly prepared material and whether the new annotations provide useful supervision for adaptation.

Table 8.5: Lemmatization accuracy on the newly annotated OCS test set. Best result in bold.

Training setting	**System**	**Accuracy (%)**
Pretrained	Stanza	15.56
Pretrained	UDPipe 2	**16.27**
New data	Dictionary baseline	37.67
New data	Stanza	51.42
New data	OLDSLAVICLEMMA	**62.91**
Combined data	Stanza	45.92
Combined data	OLDSLAVICLEMMA	**70.90**

Table 8.5 shows that the newly annotated OCS material constitutes a non-trivial benchmark for lemmatization. The off-the-shelf systems generalize poorly to this test set: pretrained Stanza and pretrained UDPipe 2 reach only 15.56% and 16.27% accuracy, respectively. A simple dictionary baseline performs better, reaching 37.67%, which indicates that some degree of lexical overlap is still useful, but this remains clearly below the best retrained neural settings. When trained on the newly annotated data, OLDSLAVICLEMMA obtains the strongest result, reaching 62.91%, compared with 51.42% for Stanza. The best overall result is obtained by OLDSLAVICLEMMA retrained on the combined data, which reaches 70.90%. This suggests that the new annotations provide task-relevant supervision that is not already captured by existing pretrained OCS models.

These results are useful for the thesis for two reasons. First, they provide direct evidence that the annotation effort is worthwhile as an evaluation resource: if the dataset were redundant with previously available OCS material, off-the-shelf systems would be expected to

perform much better. Second, they show that newly annotated in-domain material can materially improve lemmatization performance when used for retraining, especially with OldSlavicLemma. The improvement obtained by retraining OldSlavicLemma on the combined data further suggests that the new annotations complement previously available OCS resources.

8.3.2 OOV Analysis on the Newly Annotated OCS Material

To better understand why the newly annotated benchmark is difficult, we also examine out-of-vocabulary (OOV) behavior with respect to each system's own training vocabulary. This analysis is especially relevant for historical lemmatization, where orthographic variation, sparse supervision, and source mismatch frequently cause errors on unseen forms.

Table 8.6: Model-centered OOV analysis on the newly annotated OCS test set under gold tokenization. The test set contains 2,931 tokens. OOV is defined with respect to each system's own training vocabulary. Best OOV accuracy is shown in bold.

Training setting	System	Train tokens	OOV tokens	OOV %	Correct OOV	OOV Acc.
Pretrained	Stanza	159,710	2,617	89.29	199	7.60
Pretrained	UDPipe 2	159,710	2,617	89.29	199	7.60
New data	Stanza	23,768	1,854	63.25	448	24.16
New data	OldSlavicLemma	23,768	1,854	63.25	817	**44.07**
Combined data	Stanza	183,478	1,778	60.66	252	14.17
Combined data	OldSlavicLemma	183,478	1,778	60.66	1,063	**59.79**

The OOV analysis in Table 8.6 clarifies the difficulty of the new benchmark. For both pretrained systems, 89.29% of the test tokens are OOV with respect to their own training data, and OOV accuracy is only 7.60%. This helps explain the very low overall accuracy reported above. Retraining the stanza on the newly annotated material substantially improves OOV handling: the retrained Stanza model reaches 24.16% OOV accuracy, clearly above both pretrained systems and above the combined-data retrained Stanza model but far below combined-data OldSlavicLemma 59.79%. This indicates that the new dataset contributes not only additional test material, but also lexical and orthographic evidence that is directly useful for adaptation to unseen forms. OldSlavicLemma achieves the strongest OOV performance, suggesting that its neural architecture improves generalization to unseen forms in the newly annotated OCS material.

At the same time, the OOV results also show that unseen forms remain the main bottleneck. Even after retraining, OOV accuracy remains much lower than would be desirable for robust historical lemmatization. This supports one of the central claims of the thesis: in Early Slavic NLP, performance is strongly conditioned on orthographic variation, dataset mismatch, and the ability of a model to generalize beyond memorized lexical evidence.

Overall, these additional results strengthen the interpretation that the newly annotated OCS material is a meaningful contribution to the evaluation landscape of Early Slavic lemmatization, precisely because it exposes generalization limits that are not visible from standard benchmark data alone.

For this benchmark, we compare not only off-the-shelf systems but also two retrained Stanza and OldSlavicLemma settings: one trained on the newly annotated OCS material only, and one trained on a combined corpus formed by merging the new annotations with the existing UD v2.12 OCS training data. This allows us to distinguish in-domain adaptation from broader combined-resource training.

8.4 Ablation Studies

We conduct two ablation suites to isolate which design decisions contribute most to accuracy: (i) optimizer choice and (ii) the attention components (CLMA and EDCA) layered over the BiLSTM encoder–decoder backbone.

8.4.1 Optimizer Ablation: Adam vs. Lion

As shown in Table 8.7, Lion [20] reliably outperforms Adam [51] on all Early Slavic corpora, with the performance improvement being particularly significant on the most difficult OES scenario. This suggests that optimization behavior plays a significant role in low-resource historical lemmatization, the high OOV rate needs to acquire strong char-level transformations under sparse supervision.

Table 8.7: Optimizer ablation on Early Slavic treebanks (%).

Language	Treebank	Adam	Lion
Old Church Slavonic	PROIEL	95.51	96.38
Old East Slavic	Birchbark	72.01	76.55
Old East Slavic	RNC	79.30	82.89
Old East Slavic	TOROT	93.39	94.52

8.4.2 Architectural Ablation: CLMA and EDCA

As shown in Table 8.8, the removal of both attention components causes a significant performance deterioration, which supports the fact that stacked recurrent layers alone are insufficient for obtaining the required long-range dependencies and for coordinating context with lemma generation in these corpora. EDCA contributes the most to the performance enhancement, which is consistent with the requirement for coordinating the encoder and decoder with the entire serialized context window.

Table 8.8: Architectural ablation on Early Slavic UD treebanks (EM, %).

Language	Treebank	baseline	baseline + CLMA	baseline + EDCA	Full Model
Old Church Slavonic	PROIEL	11.67	90.98	96.33	96.38
Old East Slavic	Birchbark	2.68	72.37	76.05	76.55
Old East Slavic	RNC	21.30	76.57	82.71	82.89
Old East Slavic	TOROT	29.23	88.02	94.48	94.52

CLMA provides consistent complementary benefits through the integration of lower-level character patterns with higher-level context abstractions for improved robustness with noisy surface forms that may be less than fully informative.

8.5 Error Analysis: OOV vs. Ambiguity vs. Seen Forms

To understand remaining errors, Table 8.9 group test tokens into three categories: (i) **OOV** tokens not observed in training, (ii) **Ambiguous** tokens whose surface form is aligned to multiple lemmas in training, (iii) **Unambiguous seen** tokens aligned to exactly one lemma in training.

Category-level scores are computed with strict token-level exact match; unlike the official CoNLL-2018 scorer, gold "_" lemmas are not counted as automatically correct, causing minor effect on OES–Birchbark and OES–RNC.

Table 8.9: Error analysis by token category across treebanks UD v2.12: Our model vs Stanza and UDPipe 2. Best in bold.

Metric	OCS–PROIEL			OES–Birchbark			OES–RNC			OES–TOROT		
	Our	Stanza	UDPipe	Our	Stanza	UDPipe	Our	Stanza	UDPipe	Our	Stanza	UDPipe
OOV Total	4307	4307	4307	4527	4527	4527	7690	7690	7690	5962	5962	5962
OOV Correct	**3774**	3444	2512	**2383**	2098	1330	**4800**	4454	4003	**4824**	4340	2993
OOV Error	**533**	863	1795	**2144**	2429	3197	**2890**	3236	3687	**1138**	1622	2969
OOV Accuracy %	**87.62**	79.96	58.32	**52.64**	46.34	29.38	**62.42**	57.92	52.05	**80.91**	72.79	50.20
Ambig Total	5176	5176	5176	2580	2580	2580	1980	1980	1980	9661	9661	9661
Ambig Correct	5039	5040	**5049**	2454	2466	**2475**	1797	**1812**	1798	9403	9396	**9461**
Ambig Error	137	136	**127**	126	114	**105**	183	**168**	182	258	265	**200**
Ambig Accuracy %	97.35	97.37	**97.55**	95.12	95.58	**95.93**	90.76	**91.52**	90.81	97.33	97.26	**97.93**
Unambig Total	10666	10666	10666	2897	2897	2897	11207	11207	11207	12255	12255	12255
Unambig Correct	10607	10608	**10612**	2821	**2823**	2752	10708	**10745**	10509	12124	**12139**	12117
Unambig Error	59	58	**54**	76	**74**	145	499	**462**	698	131	**116**	138
Unambig Accuracy %	99.45	99.46	**99.49**	97.38	**97.45**	94.99	95.55	**95.88**	93.77	98.93	**99.05**	98.87
Overall EM %	**96.38**	94.75	90.19	**76.55**	73.84	65.54	**82.89**	81.48	78.12	**94.52**	92.82	88.14

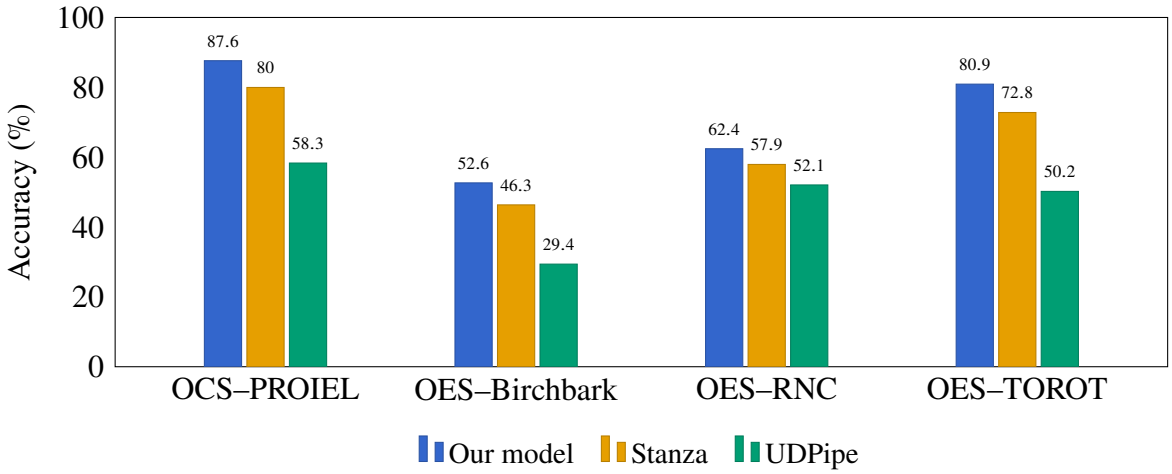

Figure 8.1: Three-model comparison across treebanks for OOV accuracy.

The breakdown shows that OOV tokens dominate errors for all systems, especially in OES datasets with high spelling variation. OldSlavicLemma achieves substantially higher OOV accuracy than Stanza and UDPipe 2, explaining most of the overall improvements, as

shown in Fig. 8.1. In contrast, performance on ambiguous and unambiguous seen forms is generally high for all methods, with smaller differences, indicating that the primary advantage of OldSlavicLemma lies in generalization under sparsity and orthographic variation rather than memorization of frequent mappings.

Chapter 9

Conclusions and Future Work

This thesis has explored lemmatization for Early Slavic texts from the perspective of end-to-end robustness, driven by two intertwined challenges. First, there is the linguistic challenge: Church Slavic has complex inflections, syncretisms, and lexeme changes, which increase the variety of forms and the ambiguity of lemmas. Second, there is the representation challenge: digital data from diverse sources has inherent instabilities in Unicode code point representation, such as PUA characters, combining characters, and font-based encodings. Such instabilities can cause spurious variability, increase the risk of OOV errors, and undermine lexical lookups. The thesis has argued that lemmatization for historical texts requires a unified view of normalization and modeling: a representation-aware preprocessing stage that stabilizes token representation, accompanied by a lemmatizer that generalizes over orthographic variability and domain shift with limited supervision.

The thesis has implemented this idea through: (i) a representation-aware normalization stage for Church Slavic texts, (ii) a transparent dictionary approach for evaluating the contribution of lexical overlap, and (iii) a dictionary-free neural lemmatizer, OldSlavicLemma, that generalizes over variability through character transduction conditioned on context. The experiments on Early Slavic UD treebanks, multilingual UD corpora, and SIGTYP 2024 historical test suite have revealed consistent improvements over strong baselines, with larger margins in OOV-rich regimes where historical variability most impacts accuracy. The central empirical finding of this thesis is that the main advantage of OldSlavicLemma does not arise from memorization of frequent mappings, but from improved generalization to unseen and orthographically variable forms, which are the dominant failure mode in historical lemmatization.

9.1 Summary of Contributions

This thesis has contributed to historical NLP for Early Slavic languages at three levels: resources, methodology, and empirical evaluation.

This thesis has proposed the design and development of the standardization layer, which is intended to address the issue of non-standard character usage, such as the use of PUA

codes, and convert them into standardized Unicode. As such, this design is explicitly corpus- and snapshot-aware. It is intended to address the representational noise, which affects the statistics of the words in the vocabulary, the rates of out-of-vocabulary words, and the quality of dictionary-based and neural approaches. As such, it confirms the contribution to the field not only in terms of the direct application to the task of lemmatization but also in terms of the workflow for the aggregation of corpora in the context of portals, editions, and Universal Dependencies-like resources.

We employed a simple dictionary-based lemmatizer, which we implemented and used as a conservative baseline to control for the impact of memorization. By adopting a most-frequent-lemma strategy for in-vocab forms and OOV pass-through, the errors can be directly related to coverage and lexical ambiguity, rather than modeling dynamics. The simple baseline served a scientific function in two ways: it allowed us to generalize beyond lexical coverage, and it offered a diagnostic window into the error profile of interest for Early Slavic texts.

The most significant modeling contribution of the thesis is the character-level encoder-decoder lemmatizer OldSlavicLemma, which utilizes a fixed context window for conditioning transduction. The model combines stacked BiLSTMs, CLMA, and EDCA. The combination allows for the decoder to utilize the entire serialized context for lemma generation instead of depending on a single representation. The model is specifically designed for generalization with limited supervision and orthography without depending on external resources. The performance of the model is found to be strong on the Early Slavic UD treebanks. The model has shown consistent improvements in the out-of-vocabulary-rich test sets, along with competitive transfer to other multilingual/historical test cases in standardized evaluation protocols.

Finally, the thesis has contributed an evaluation framework designed to make historically relevant error sources visible. The process is carried out under the gold tokenization scheme to understand the behavior of lemmatization. It is evaluated using popular baselines such as UDPipe 2 and Stanza. Category-based evaluation is also included to understand the behavior of the model with seen and unseen words (OOV), as well as in ambiguous and unambiguous regimes. Ablation studies on optimization and architecture components also helped to understand the factors behind the improvement in low-resource history settings. These components together provide a template to evaluate the historical lemmatizers with the focus being on robustness rather than the overall performance.

A further contribution of this thesis was the construction and validation of a newly annotated OCS benchmark dataset. This resource extends evaluation beyond standard UD materials and provides a more challenging testbed for robustness, adaptation, and orthographic variability. The benchmark also offers new supervised material for future Early Slavic NLP research.

9.2 Answers to the Thesis Objectives

The results presented in this thesis show that the five main objectives stated in Chapter 1 were achieved. First, the encoding-aware standardization layer demonstrated that representation-level inconsistencies such as PUA usage can be systematically detected and normalized, improving interoperability and reducing artificial variability in the data. Second, the dictionary-based baseline provided a transparent reference point for identifying the role of lexical overlap, coverage limitations, and ambiguity in Early Slavic lemmatization. Third, the proposed neural model, OldSlavicLemma, showed that character-level sequence transduction with contextual conditioning can outperform strong baselines on Early Slavic treebanks and generalize better in OOV-heavy settings. Fourth, the evaluation showed that character-level contextual lemmatization improves generalization to OOV forms, orthographically variable forms, ambiguous surface forms, and cross-domain Early Slavic material. Fifth, the newly annotated OCS benchmark confirmed the importance of newly curated material for testing robustness and adaptation beyond standard benchmark resources.

9.3 Limitations

Despite the thesis positive results, several limitations highlight areas for improvement.

The initial evaluations rely on UD treebanks and splits in a style similar to SIGTYP. This standardization of tokenization and lemmas is specific in certain ways; however, these conventions may not entirely suit the scope of Church Slavic texts and editions, in which tokenization may not always be obvious, abbreviations may have varying expansions, and lemma conventions may vary by tradition. The results reported here should thus be considered robust under benchmark conventions rather than a complete assessment of all Early Slavic editorial traditions.

A related limitation concerns annotation validation of the newly annotated OCS benchmark. Since the annotations were reviewed by a single expert linguistic annotator rather than by multiple independent annotators, no formal measure of inter-annotator agreement was calculated. Instead, quality assurance relied on expert review, manual correction, and a series of structural, encoding, lemma-field, and linguistic consistency checks. Future work should include double annotation of a representative subset of the data and report an agreement metric such as Cohen's Kappa or Krippendorff's Alpha.

The main experiments assume gold tokenization and sentence segmentation, although a supplementary raw-text-to-CoNLL-U setting is also reported using the Stanza tokenizer. This is methodologically necessary to isolate lemmatization, but it leaves open the question of end-to-end performance on raw manuscript-derived text, where segmentation errors can dominate. Although the thesis argues for a representation-aware pipeline, upstream segmentation remains an unresolved component in the full real-world workflow.

OldSlavicLemma conditions on a fixed-size window around the target token. This is a deliberate choice to balance contextual disambiguation with data efficiency, but it may be

insufficient for phenomena requiring broader discourse context (e.g., long-distance agreement cues, document-level naming conventions, or rare senses that depend on topic). The model is therefore not a general contextual language model; its context is targeted and bounded.

The proposed model is a recurrent encoder–decoder with attention rather than a pretrained Transformer. In languages or settings where unconstrained pretrained models approach near-ceiling performance, dictionary-free task training may be at a disadvantage. The multilingual UD results suggest that some languages benefit strongly from pretrained representations or from language-specific tokenization conventions, and a purely task-trained model may not close that gap without additional sources of information.

While the standardization approach is systematic, mapping PUA code points remains a practical maintenance burden in the presence of changing portals, font updates, and evolving editorial practices. The approach is robust insofar as diagnostics and versioning are maintained, but it does not eliminate the underlying socio-technical issue that PUA practices can change over time.

9.4 Future Directions

The results and limitations suggest several high-impact directions for extending this work, both scientifically and practically.

One natural extension is to use a form of joint or multi-task learning for lemma generation and morphosyntactic tagging, which utilizes the cross-constraints between these two sub-tasks. This may improve homograph and syncretic form disambiguation and could also potentially reduce reliance on local context windows by providing explicit grammatical information. Such a framework is also consistent with the UD pipeline approach but with a character-level transduction backbone.

The gap between constrained character-level models and unconstrained transformer-based approaches in some languages suggests the need for exploring hybrid approaches. One such approach is to use domain adaptive pretraining on existing Church Slavic and other Slavic language data (even if noisy), and another is to use parameter-efficient fine-tuning (e.g., adapters or low-rank updates) to combat overfitting on small gold data. A key methodological concern is how to do language-specific adaptation without reintroducing instability in the representation space due to tokenization and encoding mismatches.

A promising avenue for further research is to explore the integration of lemmatization with segmentation under noisy conditions. This includes investigating how normalization interacts with tokenization heuristics and whether character-level models can be trained to accommodate segmentation shifts, such as those involving punctuation, abbreviation markers, or ambiguous boundary characters. A more thorough evaluation at the end-to-end level will be more representative of actual editorial processes and will provide a more complete assessment of usability.

Although PUA to Unicode mapping alleviates a significant problem, historical texts also

differ in diacritical marks, rules for abbreviation expansion, and cases of mixed-script substitutions, not all of which can be directly accounted for in terms of PUA-Unicode mappings. Further research should involve formalizing additional normalization levels or multiple normalization profiles and testing them under controlled conditions to separate linguistic variation from editorial processes.

This is because the Early Slavic setup is naturally conducive to the design and conducting of transfer experiments. Such studies involve training on one redaction or genre and testing on another, with the metadata (when available) offering direct control. They promise to shed light on the learnability of certain aspects of orthographic variation with character-based modeling and domains where adaptation support remains a requirement.

A further promising direction is to extend the annotation process to the realm of active learning. It is possible to leverage the model's errors to streamline the annotation process and reduce costs. It can also help to expedite the creation of high-quality data for less represented sources. Yet another direction is to leverage the model to automatically derive interpretable data, which can be useful for supporting the work of philologists.

Future work should also include the extension of gold-standard annotation to additional Early Slavic sources, genres, and redactions. Expanding the benchmark in a controlled and interoperable way would make it possible to study diachronic variation, cross-redaction transfer, and genre-specific robustness more systematically.

Closing Remarks

Early Slavic lemmatization stands at the intersection of low-resource learning and digital philology, where progress depends both on representational stability and on careful evaluation alongside advances in model architecture. This thesis showed that a dictionary-free character-level model conditioned on context is not only competitive with popular baselines in Early Slavic but also extends its scope to a wider historical period if encoding variability is made explicit. The general significance of this research is that historical NLP should consider normalization and modeling as a unified scientific problem. The potential of research that unifies representation-aware preprocessing and improved contextualization and end-to-end evaluation is significant for increasing the reliability of large-scale processing of heterogeneous historical data.

Bibliography

[1] Muhammad Abdul-Mageed, Mona Diab, and Sandra Kübler. Samar: Subjectivity and sentiment analysis for arabic social media. *Computer Speech & Language*, 28(1):20–37, 2014.

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[3] Ilia Afanasev. A hybrid lemmatiser for old church slavonic. *Higher School of Economics Research Paper No. WP BRP*, 106, 2021.

[4] Ilia Afanasev and Olga Lyashevskaya. String similarity measures for evaluating the lemmatisation in old church slavonic. *studies in digital linguistics*, page 13, 2025.

[5] Irfan Ali, Liliana Lo Presti, Igor Spano, and Marco La Cascia. Abbie: Attention-based bi-encoders for predicting where to split compound sanskrit words. *ICAART*, 2:334–344, 2025.

[6] Irfan Ali, Liliana Lo Presti, Igor Spano', and Marco La Cascia. A unified attention-based model for segmenting compound words in sanskrit. In *International Conference on Document Analysis and Recognition*, pages 60–76. Springer, 2025.

[7] A. Andreev, Y. Shardt, and N. Simmons. Proposal to encode some additional symbols used in church slavonic text, 2015b. L2/15-173 (Revision 2), http://www.unicode.org/L2/L2015/15173r2-add-typicon.pdf.

[8] Aleksandr Andreev, Yuri Shardt, and Nikita Simmons. Church slavonic typography in unicode. *Unicode Technical*, 2015.

[9] Monika Arora and Vineet Kansal. Character level embedding with deep convolutional neural network for text normalization of unstructured data for twitter sentiment analysis. *Social Network Analysis and Mining*, 9(1):12, 2019.

[10] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[11] Vimala Balakrishnan and Ethel Lloyd-Yemoh. Stemming and lemmatization: A comparison of retrieval performances. *Lecture notes on software engineering*, 2(3):262, 2014.

[12] Jatayu Baxi and Brijesh Bhatt. A bidirectional lstm-based morphological analyzer for gujarati. *Natural Language Processing*, 31(2):198–214, 2025.

[13] Toms Bergmanis and Sharon Goldwater. Context sensitive neural lemmatization with lematus. In *16th annual conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1391–1400. Association for Computational Linguistics (ACL), 2018.

[14] Toms Bergmanis, Katharina von der Wense, Hinrich Schütze, and Sharon Goldwater. Training data augmentation for low-resource morphological inflection. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 31–39, 2017.

[15] David J Birnbaum, Ralph Cleminson, Sebastian Kempgen, and Kiril Ribarov. White paper on character set standardization for early cyrillic writing after unicode 5.1. 2008.

[16] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146, 2017.

[17] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[18] Maria Cassese, Giovanni Puccetti, Marianna Napolitano, and Andrea Esuli. Diacu: A dataset for the diachronic analysis of church slavonic. In *Proceedings of the 10th Workshop on Slavic Natural Language Processing (Slavic NLP 2025)*, pages 101–107, 2025.

[19] Abhisek Chakrabarty, Onkar Arun Pandit, and Utpal Garain. Context sensitive lemmatization using two successive bidirectional gated recurrent networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1481–1491, 2017.

[20] Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, et al. Symbolic discovery of optimization algorithms. *Advances in neural information processing systems*, 36:49205–49233, 2023.

[21] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics.

[22] Grzegorz Chrupała. Simple data-driven context-sensitive lemmatization. *Procesamiento del lenguaje natural, nº 37 (sept. 2006), pp. 121-127*, 2006.

[23] Grzegorz Chrupała, Georgiana Dinu, and Josef Van Genabith. Learning morphology with morfette. 2008.

[24] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 8440–8451, 2020.

[25] Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sandra Kübler, David Yarowsky, Jason Eisner, et al. Conll-sigmorphon 2017 shared task: Universal morphological reinflection in 52 languages. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 1–30, 2017.

[26] Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. The sigmorphon 2016 shared task—morphological reinflection. In *Proceedings of the 14th SIGMORPHON workshop on computational research in phonetics, phonology, and morphology*, pages 10–22, 2016.

[27] Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. Universal Dependencies. *Computational Linguistics*, 47(2):255–308, June 2021.

[28] Oksana Dereza, Adrian Doyle, Priya Rani, Atul Kr Ojha, Pádraic Moran, and John Philip McCrae. Findings of the sigtyp 2024 shared task on word embedding evaluation for ancient and historical languages. In *Proceedings of the 6th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pages 160–172, 2024.

[29] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

[30] Aleksei Dorkin and Kairit Sirts. Tartunlp@ sigtyp 2024 shared task: Adapting xlm-roberta for ancient and historical languages. In *Proceedings of the 6th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pages 120–130, 2024.

[31] Sašo Džeroski and Tomaž Erjavec. Learning to lemmatise slovene words. In *International Conference on Learning Language in Logic*, pages 69–88. Springer, 1999.

[32] Hanne Martine Eckhoff and Aleksandrs Berdicevskis. Linguistics vs. digital editions: The tromsø old russian and ocs treebank. 2015.

[33] Tomaž Erjavec and SASČO DŽEROSKI. Machine learning of morphosyntactic structure: Lemmatizing unknown slovene words. *Applied artificial intelligence*, 18(1):17–41, 2004.

[34] Yoav Goldberg and Omer Levy. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014.

[35] Sharon Goldwater and David McClosky. Improving statistical mt through morphological analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 676–683, 2005.

[36] Naman Goyal, Jingfei Du, Myle Ott, Giri Anantharaman, and Alexis Conneau. Larger-scale transformers for multilingual masked language modeling. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 29–33, 2021.

[37] Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

[38] Handré J Groenewald and Gerhard B Van Huyssteen. Outomatiese lemma-identifisering vir afrikaans. *Literator: Journal of Literary Criticism, Comparative Linguistics and Literary Studies*, 29(1):65–91, 2008.

[39] Michael L Hann. Principles of automatic lemmatisation. *ITL Review of Applied Linguistics*, 1974.

[40] Dag TT Haug and Marius Jøhndal. Creating a parallel treebank of the old indo-european bible translations. In *Proceedings of the second workshop on language technology for cultural heritage data (LaTeCH 2008)*, pages 27–34. Prague, 2008.

[41] Johannes Heinecke. Udparse@ sigtyp 2024 shared task: Modern language models for historical languages. In *Proceedings of the 6th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pages 142–150, 2024.

[42] Martin Hetényi and Peter Ivanič. The contribution of ss. cyril and methodius to culture and religion. *Religions*, 12(6):417, 2021.

[43] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[44] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR, 2019.

[45] Aiusha Vellintihun Hujon, Thoudam Doren Singh, and Khwairakpam Amitab. Neural machine translation systems for english to khasi: A case study of an austroasiatic language. *Expert Systems with Applications*, 238:121813, 2024.

[46] Anjali Ganesh Jivani et al. A comparative study of stemming algorithms. *Int. J. Comp. Tech. Appl*, 2(6):1930–1938, 2011.

[47] Nabam Kakum, Sahinur Rahman Laskar, Koj Sambyo, and Partha Pakray. Neural machine translation for limited resources english-nyishi pair. *Sādhanā*, 48(4):237, 2023.

[48] Jenna Kanerva, Filip Ginter, and Tapio Salakoski. Universal lemmatizer: A sequence-to-sequence model for lemmatizing universal dependencies treebanks. *Natural Language Engineering*, 27(5):545–574, 2021.

[49] Jakub Kanis and Lucie Skorkovská. Comparison of different lemmatization approaches through the means of information retrieval performance. In *International Conference on Text, Speech and Dialogue*, pages 93–100. Springer, 2010.

[50] Sebastian Kempgen. Unicode 5.1, old church slavonic, remaining problems–and solutions, including opentype features. In *Slovo: Towards a Digital Library of South Slavic Manuscripts; Proceedings of the International Conference, 21–26 February 2008, Sofia, Bulgaria*. Otto-Friedrich-Universität, 2008.

[51] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[52] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, pages 177–180, 2007.

[53] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings*

of the 2018 conference on empirical methods in natural language processing: System demonstrations, pages 66–71, 2018.

[54] Avinash Kumar, Vishnu Teja Narapareddy, Veerubhotla Aditya Srikanth, Aruna Malapati, and Lalita Bhanu Murthy Neti. Sarcasm detection using multi-head attention based bidirectional lstm. *Ieee Access*, 8:6388–6397, 2020.

[55] Piroska Lendvai, Uwe Reichel, Anna Jouravel, Achim Rabus, and Elena Renje. Domain-adapting bert for attributing manuscript, century and region in pre-modern slavic texts. In *Proceedings of the 4th Workshop on Computational Approaches to Historical Language Change*, pages 15–21, 2023.

[56] Piroska Lendvai, Uwe Reichel, Anna Jouravel, Achim Rabus, and Elena Renje. Retrieval of parallelizable texts across church slavic variants. In *Proceedings of the 12th Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 105–114, 2025.

[57] Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, et al. Xglue: A new benchmark dataset for cross-lingual pre-training, understanding and generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6008–6018, 2020.

[58] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[59] Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In Lluís Màrquez, Chris Callison-Burch, and Jian Su, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[60] C MacRobert. Old church slavonic. *Encyclopedia of Language and Linguistics (Second Edition)*, 2006.

[61] Enrique Manjavacas, Ákos Kádár, and Mike Kestemont. Improving lemmatization of non-standard languages with joint learning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1493–1503, 2019.

[62] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[63] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

[64] Lester James V Miranda. Allen institute for ai@ sigtyp 2024 shared task on word embedding evaluation for ancient and historical languages. In *Proceedings of the 6th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pages 151–159, 2024.

[65] Aiom Minnette Mitri, Eusebius Lawai Lyngdoh, Sunita Warjri, Goutam Saha, Saralin A Lyngdoh, and Arnab Kumar Maji. Probing a pretrained roberta on khasi language for pos tagging. *Natural Language Processing*, 31(2):230–249, 2025.

[66] Thomas Müller, Ryan Cotterell, Alexander Fraser, and Hinrich Schütze. Joint lemmatization and morphological tagging with lemming. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2268–2274, 2015.

[67] Marianna Napolitano, Usman Nawaz, et al. From characters to meaning: Challenges and limitations in the semantic analysis of church slavonic texts. 2025.

[68] Usman Nawaz, Liliana Lo Presti, Marianna Napolitano, and Marco La Cascia. Automatic lemmatization of old church slavonic language using a novel dictionary-based approach. In *International Workshop on Document Analysis Systems*, pages 408–421. Springer, 2024.

[69] Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, 2016.

[70] Morrel VL Nunsanga, Partha Pakray, C Lallawmsanga, and L Singh. Part-of-speech tagging for mizo language using conditional random field. *Computación y Sistemas*, 25(4):803–812, 2021.

[71] Robert Östling and Johannes Bjerva. Su-rug at the conll-sigmorphon 2017 shared task: Morphological inflection with attentional sequence-to-sequence models. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 110–113, 2017.

[72] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[73] Partha Pakray, Alexander Gelbukh, and Sivaji Bandyopadhyay. Natural language processing applications for low-resource languages. *Natural Language Processing*, 31(2):183–197, 2025.

[74] Hariom A Pandya and Brijesh S Bhatt. Does learning from language family help? a case study on a low-resource question-answering task. *Natural Language Processing*, 31(2):375–392, 2025.

[75] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[76] Michael Piotrowski. *Natural language processing for historical texts*. Morgan & Claypool Publishers, 2012.

[77] Telmo Pires, Eva Schlinger, and Dan Garrette. How multilingual is multilingual bert? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 4996–5001, 2019.

[78] Peng Qi, Timothy Dozat, Yuhao Zhang, and Christopher D Manning. Universal dependency parsing from scratch. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual parsing from raw text to universal dependencies*, pages 160–170, 2018.

[79] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, 2020.

[80] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[81] Frederick Riemenschneider and Kevin Krahn. Heidelberg-boston@ sigtyp 2024 shared task: Enhancing low-resource language analysis with character-aware hierarchical transformers. In *Proceedings of the 6th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pages 131–141, 2024.

[82] Helmut Schmid, Arne Fitschen, and Ulrich Heid. Smor: A german computational morphology covering derivation, composition and inflection. In *LREC*, pages 1–263. Lisbon, 2004.

[83] Carsten Schnober, Steffen Eger, Erik-Lân Do Dinh, and Iryna Gurevych. Still not there? comparing traditional sequence-to-sequence models to encoder-decoder neural networks on monotone string translation tasks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1703–1714, 2016.

[84] Abigail See, Peter J Liu, and Christopher D Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, 2017.

[85] 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 (volume 1: long papers)*, pages 1715–1725, 2016.

[86] Xinying Song, Alex Salcianu, Yang Song, Dave Dopson, and Denny Zhou. Fast wordpiece tokenization. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 2089–2103, 2021.

[87] Milan Straka. Udpipe 2.0 prototype at conll 2018 ud shared task. In *Proceedings of the CoNLL 2018 shared task: Multilingual parsing from raw text to universal dependencies*, pages 197–207, 2018.

[88] Milan Straka, Jan Hajic, and Jana Straková. Udpipe: trainable pipeline for processing conll-u files performing tokenization, morphological analysis, pos tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4290–4297, 2016.

[89] Milan Straka and Jana Straková. Tokenizing, pos tagging, lemmatizing and parsing ud 2.0 with udpipe. In *Proceedings of the CoNLL 2017 shared task: Multilingual parsing from raw text to universal dependencies*, pages 88–99, 2017.

[90] Milan Straka, Jana Straková, and Jan Hajic. Udpipe at sigmorphon 2019: Contextualized embeddings, regularization with morphological categories, corpora merging. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 95–103, 2019.

[91] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

[92] TEI Consortium. Tei: Guidelines for electronic text encoding and interchange, 2026. Revision 358d2e48e.

[93] The Unicode Consortium. *The Unicode Standard, Version 16.0.0*. The Unicode Consortium, South San Francisco, CA, 2024.

[94] The Unicode Consortium. *The Unicode Standard, Version 17.0.0*. The Unicode Consortium, South San Francisco, 2025.

[95] Francielle Vargas, Isabelle Carvalho, Thiago AS Pardo, and Fabrício Benevenuto. Context-aware and expert data resources for brazilian portuguese hate speech detection. *Natural Language Processing*, 31(2):435–456, 2025.

[96] Francielle Vargas, Wolfgang Schmeisser-Nieto, Zohar Rabinovich, Thiago AS Pardo, and Fabrício Benevenuto. Discourse annotation guideline for low-resource languages. *Natural Language Processing*, 31(2):700–743, 2025.

[97] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[98] Katharina von der Wense, Ryan Cotterell, and Hinrich Schütze. Neural multi-source morphological reinflection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 514–524, 2017.

[99] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32, 2019.

[100] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP*, pages 353–355, 2018.

[101] Barack W Wanjawa, Lilian DA Wanzare, Florence Indede, Owen McOnyango, Lawrence Muchemi, and Edward Ombui. Kenswquad—a question answering dataset for swahili low-resource language. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(4):1–20, 2023.

[102] Sunita Warjri, Partha Pakray, Saralin A Lyngdoh, and Arnab Kumar Maji. Part-of-speech (pos) tagging using deep learning-based approaches on the designed khasi pos corpus. *Transactions on Asian and Low-Resource Language Information Processing*, 21(3):1–24, 2021.

[103] Peter Willett. The porter stemming algorithm: then and now. *Program*, 40(3):219–223, 2006.

[104] Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.

[105] Imad Zeroual and Abdelhak Lakhouaja. Arabic information retrieval: Stemming or lemmatization? In *2017 Intelligent Systems and Computer Vision (ISCV)*, pages 1–6. IEEE, 2017.

[106] He Zhou, Daniel Dakota, and Sandra Kübler. Cross-lingual dependency parsing for a language with a unique script. *Natural Language Processing*, 31(2):277–305, 2025.