ADDIS ABABA UNIVERSITY
ADDIS ABABA INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

ETHIO-SEMITIC PROTO-LANGUAGE RECONSTRUCTION WITH IN-CONTEXT LEARNING AND LSTM ENCODE-DECODE MODEL

BY
ELLENI SISAY

ADVISORS
Dr. FITSUM ASSAMNEW
HELLINA HAILU NIGATU

A thesis submitted to the School of Electrical and Computer Engineering in partial fulfillment of the requirements for the Degree of Master of Science in Computer Engineering

DECEMBER, 2024
ADDIS ABABA, ETHIOPIA

ADDIS ABABA UNIVERSITY

ADDIS ABABA INSTITUTE OF TECHNOLOGY

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

The undersigned have examined the thesis titled:

ANCESTOR LANGUAGE RECONSTRUCTION OF ETHIO-SEMITIC LANGUAGES

BY

ELLENI SISAY

Approval by Boards of Examiners

Dr. Bisrat Derebssa	_____	_____
Dean, SECE, AAiT	Date	Signature
Dr. Fitsum Assamnew	_____	_____
Advisor	Date	Signature
Hellina Hailu Nigatu	_____	_____
Advisor	Date	Signature
Dr. Bisrat Derebssa	_____	_____
External Examiner	Date	Signature
Dr. Menore Tekeba	_____	_____
Internal Examiner	Date	Signature

Declaration

I, Elleni Sisay Temesgen, declare that this thesis is my original work. All sources of information in this study have been appropriately acknowledged. I further confirm that this thesis has not been submitted either in part or in full for any other requirements to any other learning institution.

Student Name: Elleni Sisay Temesgen

Signature: _____

Date: _____

DECEMBER, 2024

Acknowledgments

First and foremost, I want to express my heartfelt gratitude to the almighty GOD for the countless blessings. His guidance, protection, and love have been my constant support throughout my life. After GOD I thank His mother the Virgin Mary for her guidance, protection, and love helping me achieve everything I have today.

I also want to express my sincere gratitude to my advisors Dr. Firstum Assamnew and Hellina Hailu Nigatu (PhD candidate) for their support and advice throughout my research. Their guidance, comments, and encouragement helped me to achieve this research.

I would like to thank my father Sisay Temesgen, my mom and my grandmother, my family as a whole for their unwavering support, encouragement and love throughout the challenges of my academic and personal journey.Their belief in me and continuous encouragement have been a source of strength and motivation, and I am profoundly grateful to have them as a family.

I would also like to express my heartfelt thanks to all my fellow Computer Engineering MSc lab students at AAiT. Their support and encouragement throughout the entire process have been invaluable. I have learned so much from them, and I am grateful for the opportunity to have worked and studied alongside such dedicated individuals.

Lastly, I extend my gratitude to the 6 Kilo Linguistic Department, especially Dr. Desalegn Hagos for his support in building the golden dataset. His assistance has been essential in the progress of my research.

Abstract

As language evolve, it change and words obtain new meanings and lose old ones, making their reconstruction a critical area of study. Proto-EthioSemitic languages, in particular remain underexplored despite their cultural and historical significance. This research investigates Historical Language Reconstruction (HLR) for Proto-EthioSemitic languages in word level, focusing on two core objectives: cognate identification and proto-word reconstruction. A three-way dictionary was used to compile a dataset of 14,100 semantically related words from Amharic, Ge'ez, and Tigrinya. Linguists manually identified a golden data set with 74 cognate pairs from the Swadesh list concept translated into the three languages of interest and reconstructed proto-forms, while using automated methods (SCA and LexStat) extracted an additional 1,847 cognates from the dataset, significantly enhancing scale. Building on these results, synthetic proto-forms were generated using in-context learning with GPT-4o, based on its performance of achieving a reconstruction accuracy of 85% when evaluated against the golden data. Furthermore, an LSTM-based encode-decode model was trained on the generated data to predict proto-forms from cognates, achieving a prediction accuracy of 91% and an average edit distance of 0.21. This work establishes a foundation for reconstructing ancestral languages within the Afro-Semitic family by integrating linguistic expertise, automated cognate extraction tools, and state-of-the-art large language models. The findings underscore the potential of interdisciplinary approaches in preserving and understanding linguistic heritage, with implications for future studies in historical linguistics and language preservation.

Keywords: Cognates, Proto-word, In-context learning, GPT 4o, LSTM based encode-decode.

Table of Contents

List of Figures

List of Tables

List of Acronyms

API Application Programming Interface

BERT Bidirectional Encoder Representations from Transformers

CCM Consonant-Class-Matching

CNN Convolutional Neural Networks

DNN Deep Neural Network

FNN Feed-forward Neural Networks

GPT Generative Pre-trained Transformer

GRU Gated Recurrent Unit

HLR Historical Language Reconstruction

HMM Hidden Markov Model

ICL In-Context Learning

IPA International Phonetic Alphabet

LLM Large Language Model

LSSP Language-specific Sequence Similarity Partitioning

LSTM Long Short-Term Memory

ML Machine Learning

MLP Multilayer Perceptron

MT Machine Translation

NLP Natural Language Processing

NMT Neural Machine Translation

PSSP phenotypic Sequence Similarity Partitioning

RNN Recurrent Neural Network

RNNs Recurrent Neural Networks

SCA Sound Class Algorithm

UPGMA Unweighted Pair Group Method with Arithmetic mean

VAR Variational Autoencoder

Chapter 1

Introduction

Language is a primary tool for communication allowing people to convey message through sounds, symbols, or gestures. Over time, languages change and words obtain new meanings and lose old ones. As it changes, the parent language splits and in some cases, is replaced by many daughter languages. This process results in the formation of language families, which are groups of languages that share a common ancestral language, known as the proto-language. The parent word of the daughter languages words is called proto-word [1, 2].

Historical linguistics is the branch of linguistics that studies the development and evolution of languages over time. It focuses on understanding how languages change in their phonetics. In historical linguistics, the method that aims at deducing the features of a common ancestral language from which a group of related languages has evolved is called Proto-language reconstruction. In addition, Proto-word reconstruction consists of recreating the words in an ancient language from its modern daughter languages. This process involves comparing similarities and differences among languages within a family to identify consistent patterns of sound changes in cognate words [3, 4, 5].

Cognate words are words in different languages that share common etymological origins [6]. They exhibit both phonetic and semantic resemblance due to deriving from a common ancestral language. By analyzing these patterns, linguists can infer and reconstruct the phonology, morphology, and syntax of the proto-language, despite it not being directly attested in written records. These reconstructed elements are known as proto-forms [7, 8].

Computational methods for proto-word reconstruction, particularly machine learning approaches have become increasingly popular in recent years and shown promising results[9, 10]. However, these methods have limitations as they rely heavily on access to large volumes of data, which presents challenges for historical linguists who study lesser-known language families as they primarily rely on data sets[10].

The Ethio-Semitic language family is primarily spoken in Ethiopia and parts of Eritrea. This family includes languages such as Geez, Tigrinya, and Amharic [11]. Although extensive research has been conducted to reconstruct the proto-word of other language families, to our knowledge no such research has been done on the Ethio-Semitic language family. Additionally, there is a lack of data set, making it one of the less-studied language families [12, 13].

1.1 Problem Statement

The reconstruction of historical ancestor languages is a crucial task in the field of linguistics, as it provides insights into the evolution of human language and cultural history. Linguists perform manual reconstruction of ancestor languages by identifying patterns of sound changes in cognates. Recent advancements in computational linguistics have shown promising results in reconstructing proto-languages for various language families. This has been achieved through the use of machine translation architectures such as sequence-to-sequence Gated Recurrent Unit Gated Recurrent Unit (GRU) and transformer based models. These methods excel at processing large corpus of linguistic data, identifying patterns, and predicting tasks that traditional methods often struggle to achieve[9, 10].

Despite recent successes in computational linguistics, several challenges persist when applying these methodologies to the reconstruction of the Proto-Word of Ethiopian Semitic languages. Ethiopian Semitic languages include languages such as Ge'ez, Tigrinya, and Amharic[12]. These languages exhibit distinct features in sentence structure (syntax), word structure (morphology), sound structure (phonology), and vocabulary (lexicon), remain understudied, making it difficult to fully understand their evolution[13]. Transformer based model, which require large linguistic datasets are not ideal for the less-resourced Ethio-Semitic language family, as these datasets are unavailable, preventing the models from fully capturing the unique phonological characteristics of such languages[10]. Similarly, GRU-based NMT model face challenges with noisy and uncertain data, limiting their ability to recognize pattern recognition and reconstruct languages with few historical records or complex dialectal variations[9, 10]. To address these challenges, our approach involves using Large Language Models (LLMs) to generate synthetic data, which aims to overcome the limitations of dataset availability in Ethiopian Semitic languages. LLMs have demonstrated impressive performance in natural language understanding [14] and generation [15] but have not yet been applied to ancestor language reconstruction. By generating a large dataset, the objective is to improve the training of encode-decode models and improve the accuracy of the Proto-Word reconstruction. To our knowledge, no efforts have been made to reconstruct proto-words from Ethiopian Semitic languages at the word level. Furthermore, no cognate data set is available, creating a significant gap in the field of historical linguistics. To address these issues, this research attempts to answer the following research questions:

RQ1 How effective are large language models (LLMs) at generating high-quality synthetic cognate datasets for the Ethio-Semitic language family, particularly when using in-context learning to address data scarcity and phonological uniqueness?

RQ2 To what extent can an LSTM-based encode-decode model, trained on both generated synthetic data and existing linguistic data, accurately reconstruct proto-words of Ethio-Semitic languages, compared to existing GRU and transformer models?

1.2 Objectives

1.2.1 General Objective

This research seeks to bridge the gap in the field of historical linguistics, as no efforts have been made to reconstruct proto-words from Ethiopian Semitic languages at the word level.

1.2.2 Specific Objectives

- To build a dataset of cognates for the three languages (Ge'ez, Tigrinya, and Amharic) through human experts and automated methods.
- To investigate if In-context learning using Large Language Models can be used to generate a synthetic proto-word data set for the historical word Reconstruction task.
- To evaluate existing methodologies for reconstructing the proto-form of Ethiopian Semitic languages.
- To design Long Short-Term Memory (LSTM) based encode-decode model and evaluate how it enhances the performance of Ethio-Semitic family language reconstruction.

1.3 Contribution

In this work, there are three main contributions:

- Build a detailed cognate set words dataset for the Ethio-Semitic languages: Amharic, Ge'ez, and Tigriyna. We compiled this dataset using phonetic similarity measures, alignment algorithms, and expert validation. It is a valuable resource for comparative linguistic analysis and historical language reconstruction.
- Use Generative Pre-trained Transformer (GPT)-4o LLM to generate a synthetic Proto-EthioSemetic words using In-context learning and use the generated data to train Recurrent Neural Network (RNN) and Transformer.
- Built LSTM-based encode-decode model and evaluated its efficiency on the Ethio-Semitic language family reconstruction.

1.4 Scope and Limitation

The scope of this research focuses on exploring and adapting machine translation architectures for reconstruction of Ethiopian Semitic languages Proto-word and In-context learning for for synthetic data generation. This includes tasks such as collecting and annotating data, cognate identification, refining prompts for context-based learning, and designing, evaluation frameworks that best fit linguistic features of these languages.

The limitations of this research are:

- The reconstruction focuses only on three Ethio-Semitic languages Ge'ez, Amharic, and Tigrinya, out of seven total languages in the family.
- Part of the dataset used in this research is constructed using automated methods, which may introduce errors. These inaccuracies can affect the performance of the proposed model, as the quality of machine-constructed data can vary and potentially impact the accuracy and reliability of the reconstruction process.
- To convert the geez words to their International Phonetic Alphabet (IPA) the Tigrinya more phonetic form is used. This is because Ge'ez and Tigrinya both belong to the same category, Northern Ethio-Semitic. Additionally, Tigrinya preserves more phonetic distinctions that are not found in Amharic but are present in Ge'ez.

1.5 Organization of the study

The remaining sections of this document are structured as follows: Chapter Two covers the fundamental concepts essential for cognate detection and proto-language reconstruction within the Ethio-Semitic language family. It includes mechanisms for identifying cognates and an overview of various machine translation architectures used in ancestor language reconstruction and large language models. Chapter Three provides an in-depth review of existing machine learning approaches to ancestor language reconstruction. Chapter Four details the research methodologies employed in this study. Chapter Five presents the experiments conducted, the results obtained, and a comparative analysis of various state-of-the-art models. Finally, Chapter Six discusses the conclusions drawn from the research and outlines potential directions for future work.

Chapter 2

Background

In this chapter a detailed theoretical overview of methods used for identifying cognates and reconstructing Proto-words are present. These approaches include both comparative and computational methods. It begins with an exploration of the Ethio-Semitic language family, followed by an in-depth discussion of cognate identification, and then followed by comparative and computational techniques used in language reconstruction.

2.1 Language

Language is used as a system of communication through symbols like words and gestures to understand meaning. It is a complex and unique human ability that enables individuals to express their thoughts, feelings, ideas, and intentions, as well as to understand and interpret the messages of others. Language plays a crucial role in human evolution, culture, and identity [16, 1, 2].

2.1.1 Language Evolvement

Language is always changing and there are many routes through which this change occurs. It can originate in language learning, language contact, social differentiation, and natural processes in usage [17, 18].

- Language learning: Language transforms as it is transmitted from one generation to the next.
- Language contact: Migration, conquest, and trade bring speakers of different languages into contact.
- Social differentiation: Social groups adopt distinctive norms in dress, adornment, gestures, and language.

- Natural processes in usage: Rapid or casual speech naturally leads to processes such as assimilation, dissimilation, syncope, and apocope.

There are various types of language change, which can occur at different levels. This includes phonetic (sounds), lexical (vocabulary), grammatical (structure), and semantic (meaning) changes. This research focuses on the phonetic changes within the Ethio-Semitic language family [17].

2.1.2 Proto-Language

Over time as languages evolve the parent language splits and is replaced by multiple daughter languages. This process leads to the formation of language families, groups of languages that share a common ancestral language known as the proto-language. Linguists therefore describe the daughter languages within a language family as being genetically related [19, 3].

Researchs in historical linguistics has provided strong evidence for the existence of shared ancestral languages known as Proto-languages using carefully comparing vocabularies across different languages. Proto-languages serve as the common root for groups of languages or entire language families [20, 21, 19]. Through this detailed analysis linguists were able to identify cognates that share a common meaning and origin. Identifying cognates depends on recognizing consistent patterns of sound similarities among words in various languages. By detecting these patterns, linguists can formulate hypotheses about sound change rules and work towards reconstructing the likely phonetic forms of words in the Proto-language [22, 3].

2.2 Proto-Language Reconstruction

Reconstructing the proto-word of an ancestor language from the cognate sets of its daughter languages is known as historical ancestor language reconstruction. This process uses careful methods to uncover the phonetic, lexical, and grammatical features of these ancient languages, giving valuable insights into how languages have evolved and connected over time [23]. Language reconstruction relies heavily on the comparative method, where linguists closely examine the similarities and differences among languages to spot patterns in their development. Cognate analysis is one of the crucial parts of the reconstruction process. It involves identifying words with a common origin across different languages called cognates. By looking at sound correspondences and tracking changes in meaning linguists can piece together hypothetical vocabularies and grammatical structures of the Proto-language [24].

Traditionally reconstructing Proto-languages was a painstaking process carried out manually by linguists. It demands a lot of time and effort [25]. However, in recent times computational techniques have become increasingly crucial in this field, especially for managing large datasets and complex linguistic connections. These days, researchers use various computational methods to analyze linguistic data and uncover the features of ancestral or Proto-languages. Some common techniques include phylogenetic methods, Bayesian inference, and machine translation models, all of which play significant roles in advancing the understanding of ancestor language reconstruction.

2.3 Ethiopian Semitic Languages

Proto-IndoEuropean is considered the ancient ancestor of many languages within the Indo-European family, spanning from languages spoken in Europe to parts of Asia [26]. Similarly, Proto-Afro-Asiatic [1] is the common ancestor of the Afro-Asiatic language family [27]. This family includes languages spoken across parts of North Africa, the Horn of Africa, the Arabian Peninsula, and the Middle East. Some of its most well-known branches include Semitic languages (like Arabic, Hebrew, Amharic, and Ge'ez), Berber, Egyptian, Chadic, Cushitic, and Omotic languages [28, 29].

[1] https://en.wikipedia.org/wiki/Proto-Afroasiatic$_l anguage$

The Semitic language family includes major divisions like East Semitic (including Akkadian and Assyrian), West Semitic (including Arabic, Hebrew, and Aramaic), Central Semitic (including Arabic and Aramaic) and South Semitic (including Ethiopian languages like Amharic and Tigrinya) [29, 30].

Ethiopian Semitic languages form a subset within the Semitic family, centered primarily in Ethiopia and Eritrea. These languages include Ge'ez, Tigrinya, Tigre, Amharic, Argobba, Gurage, and Harari. Proto-Ethiopic serves as the reconstructed ancestor language for these Ethiopian Semitic languages, representing their historical and linguistic origins [30, 11].

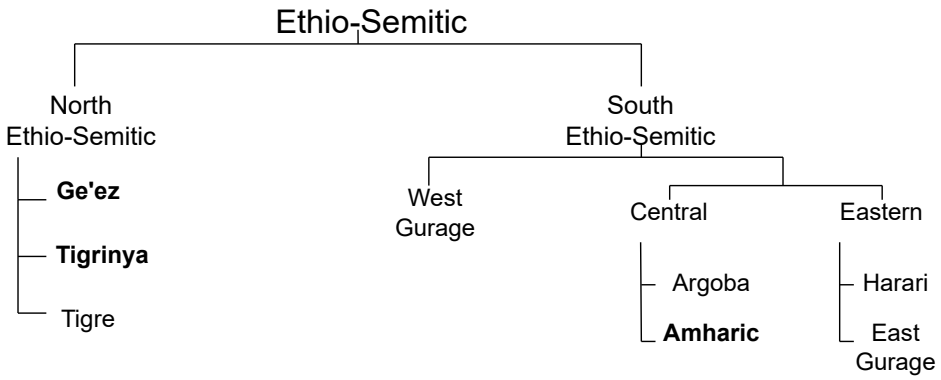

Figure 2.1: Ethio-Semitic language familly

2.3.1 Ge'ez

Ge'ez, also known as Ethiopic is an ancient Semitic language originating from present-day Eritrea and northern Ethiopia. It belongs to the North Ethiopian Semitic branch of the Afro-Asiatic language family. It holds significant historical and cultural importance as Ethiopia's classical language. Ge'ez has served as a main language for religious texts, including translations of the Christian Bible for the Ethiopian Orthodox Tewahedo Church, shaping the spiritual and cultural identity of Ethiopian and Eritrean societies [30, 11].

Linguistically, Ge'ez is distinguished by its Ge'ez script an 'Abugida' writing system where each symbol represents a consonant-vowel combination. This script has also been adapted to write other Ethiopian and Eritrean languages like Amharic and Tigrinya.

2.3.2 Amharic

Amharic is spoken not only in its home province, Amhara but over large areas of Central Ethiopia. It is one of the official languages of Ethiopia, spoken by over 33.7 million people as a first language and 25.1 million as a second language according to the Central Statistical Agency of Ethiopia. Amharic has a unique orthographic representation containing 32 consonants and 7 vowels called Amharic-Fidel. Amharic has undergone profound changes in its phonetic character: the laryngals have been reduced to h and the glottal stop and even the latter is rare now. The orthographic representation is also shared with Tigrinya, the other Semitic language of Ethiopia [30, 11].

2.3.3 Tigrinya

Tigrinya is spoken in the area largely identical to that of the old Aksumite empire. Currently, it is one of the official languages of Ethiopia and Eritrea and is spoken by 9.7 million people[2] in total across the two countries and their diasporas. Tigrinya uses the Ge'ez Script. Within the Tigrinya alphabet, there are 35 consonants and 7 vowels in the writing system [31].

2.4 International Phonetic Alphabet

International Phonetic Alphabet(IPA)[3] is a standardized system of symbols used to represent the sounds of spoken language. One of its primary goals is to assign a unique symbol to each distinct sound, known as a phoneme that distinguishes one word from another within a language. The IPA aims to accurately represent the various aspects of speech that contribute to lexical and to some extent prosodic elements in spoken language, including phones, intonation, and syllable breaks. For more nuanced speech qualities like tooth gnashing, lisping, or sounds produced while speaking an extended set of symbols may be employed beyond the basic IPA set. Translating text into IPA is useful in linguistic research, language learning, language reconstruction, and other applications where a precise representation of pronunciation is needed [32, 33].

[2] https://en.wikipedia.org/wiki/Tigrinya$_{l}anguagecite_{n}ote-E27-1$

[3] https://en.wikipedia.org/wiki/International$_{P}honetic_{A}lphabet$

As of the most recent change, there are more than 160 different IPA symbols containing 107 segmental letters, an indefinitely large number of suprasegmental letters, 44 diacritics (not counting composites), and four extra-lexical prosodic marks in the IPA. To use an IPA translator, words or text are typically entered into the converter like metaappz word by word, and then the corresponding phonetic symbols are given. For example: the Amharic word ethioplbs IPA phonetics is 'lɨbɨs'. These translations can be done using libraries [33].

2.4.1 Epitran

Epitran[4] is a library and tool designed for translating orthographic text into the IPA. It supports 61 languages including Amharic and Tigriniya. The phonetic representation of most languages varies and these languages have their own unique sound than others by considering this Epitran addressed 61 languages [34, 35].

Epitran typically focuses on representing the phonetic details of a language and allows to conversion of written text into a phonetic transcription that reflects the sounds of the spoken words. This helps particularly in linguistics, to analyze or compare the phonetic features of different languages. Epitran have specific implementations or modules for different languages, as each language may have its own orthographic conventions and pronunciation rules [35].

2.5 Cognate Identification

Cognates are words in different languages that have similar forms and meanings, due to a common linguistic origin from a shared ancestor language. Words genetically related to the same descendants from a common ancestral language are termed cognates [6]. For example, the Ge'ez word `k$^{w}$illu', the Amharic `hullu', and the Tigriniya word ``kulu'' are cognates and their reconstructed proto-word is `kullu' as ancestral Proto-EthioSemitic. Within historical linguistics assembling potential cognate forms is an essential step in the comparative method to proceed to further stages such as formulation of sound laws, reconstruction of proto-language, and phylogenetic reconstruction[37].

[4]https://github.com/dmort27/epitran/tree/master

tir-Ethi	Native Tigrinya	Tigrinya in its standard orthography using the Ge'ez script [36]
tir-Ethi-pp	Tigrinya (more phonemic)	With a more phonemic approach, where the orthography might be adapted to reflect the phonemic aspects of the language better. It possibly makes it easier to pronounce for learners or reflects spoken variations more closely(The phonetic representation of Tigriniya words) [36].
tir-Ethi-red	Tigrinya (reduced)	A reduced or simplified orthography for Tigrinya. This could involve using fewer characters or a simplified script to make reading and writing more accessible
amh-Ethi	Amharic	Amharic in its standard orthography using the Ge'ez script [36]
amh-Ethi-pp	Amharic (more phonetic)	Amharic with a more phonetic approach. The orthography here might be adapted to reflect the pronunciation more closely, which can help with learning correct pronunciation or for linguistic studies [36].
amh-Ethi-red	Amharic (reduced)	Represents a reduced or simplified orthography for Amharic. It might involve a simplified script or fewer characters, designed for ease of use in certain contexts such as language learning or informal communication [36].

Table 2.1: Epitran codes Definition

Cognate identification has been traditionally carried out by tedious manual cross-comparisons of lexica across several concepts or meanings. This often requires sufficient linguistic expertise in the languages that are being compared. Automated cognate detection attempts to minimize manual labor and will eventually assist a historical linguist to quickly produce the high-quality etymologies required. Daughter language words can be checked if they are cognates using two methods the comparative method, and the computational method.

2.5.1 Comparative Method

Cognate detection using the traditional comparative method involves an examination of lexical similarities across related languages. This method relies on the principle of regular sound correspondences, where phonetic changes occurring over time in the descendant languages are systematically compared to establish cognate relationships. Linguists typically compile cognate sets by analyzing words with similar meanings and forms in different languages within a language family. Then they apply sound correspondences and phonological rules to reconstruct the ancestral proto-word form of these cognates [25, 38].

Tracing the historical development of cognates through the comparative method, linguists gain insights into linguistic evolution and historical relationships between languages. This approach requires expertise in historical linguistics as well as access to extensive language data and specialized tools for phonological analysis and reconstruction. Despite its labor-intensive nature, the traditional comparative method remains a fundamental technique for cognate detection and is widely used in historical and comparative linguistics.

2.5.2 Computational Methods

Cognate detection using computational methods which is computer based follows traditional linguistic principles. It uses computational algorithms and statistical models to automate the process of identifying cognate words across related languages. Initially, a large dataset containing lexical items from multiple languages within a language family is compiled. These datasets are often sourced from linguistic databases, corpora, or digital repositories. Computational algorithms then compare phonetic and orthographic features of words across languages to assess their similarity. Common approaches include phonetic alignment algorithms such as the Soundex or Levenshtein distance, which measure the degree of phonetic resemblance between words. Additionally, statistical models like probabilistic graphical models or machine learning classifiers, are trained on labeled cognate datasets to predict cognate relationships based on phonological, morphological, and semantic features.

2.5.2.1 lingpy

LingPy [5] is a Python open-source library designed for computational historical linguistics and quantitative analysis of linguistic data. It provides a wide range of tools and algorithms for tasks such as phonetic alignment, cognate detection, sound correspondences, and language comparison. LingPy enables researchers to automate many aspects of linguistic analysis making it easier to process and analyze large datasets from diverse language families [39].

The cognate detection algorithms in LingPy utilize three major types: Consonant-Class-Matching (CCM), phenotypic Sequence Similarity Partitioning (PSSP), and Language-specific Sequence Similarity Partitioning (LSSP). LingPy offers four main methods for cognate detection, each with varying degrees of algorithmic sophistication and adherence to linguistic theory [39]:

[5] https://lingpy.org/

1. **Turchin [40]** (also called Consonant Class Matching(CCM) approach following [41] early idea to assume that words with two matching consonant classes would likely be cognate)) was proposed by peter[40]. In this method, the consonants of the words are converted to one of 10 possible consonant classes. The idea of consonant classes (also called sound classes) was proposed by Dolgopolsky, who stated that certain sounds occur more frequently in a correspondence relation than others and could therefore be clustered into classes of high historical similarity. In the approach by [40], two words are judged to be cognate, if they match in their first two consonant classes.

2. **Sound Class Algorithm (SCA) [42]** uses a threshold-based clustering algorithm and employs distance scores derived from the Sound-Class Based Alignment (SCA) method. This method for pairwise and multiple alignment analyses uses expanded sound class models along with detailed scoring functions as its basis. In contrast to previous alignment algorithms, the SCA algorithm takes prosodic aspects of the words into account and is also capable of aligning within morpheme boundaries, if morpheme information is available in the input data.

3. **LexStat [43]** method is based on flat Unweighted Pair Group Method with Arithmetic mean (UPGMA) clustering, but in contrast to the SCA method, it uses language-specific scoring schemes which are derived from a Monte-Carlo permutation of the data. This permutation, by which the word lists of all language pairs are shuffled in such a way that words denoting different meanings are aligned and scored, is used to derive a distribution of sound-correspondence frequencies under the assumption that both languages are not related. The permuted distribution is then compared with the attested distribution, and converted into a language-specific scoring scheme for all language pairs. Using this scoring scheme, the words in the data are aligned again, and distance scores are derived which are then used as the basis for the flat cluster algorithm.

2.6 Ancestor Language Reconstruction

2.6.1 The Comparative Methods

As outlined by [44] Reconstruction of ancestor languages is mostly done using the traditional comparative method. Its objective is to identify genetic relationships between languages while excluding instances of borrowing. Various linguistic levels such as phonology, morphology, and syntax, are considered in this process. Attention is given to the phonological forms of a standardized list of fundamental vocabulary, which includes concepts less likely to have been borrowed in this process [25].

The step by step approach of the comparative method that is used by historical linguists are [25, 38]:

1. First gather word lists from languages believed to be genetically connected based on diagnostic evidence on morphological or syntactical similarities.
2. Next, establish sets of cognates, words with shared ancestral roots.
3. By Examining these cognates, linguists pinpoint consistent sound changes between words in different languages.
4. Using these sound correspondences linguists reconstruct a common ancestral language by initially inferring proto sounds and then reconstructing protoforms of words.
5. This information is then used to construct a phylogenetic tree, illustrating the evolutionary relationships among languages. For instance, if language B exhibits all the innovations of language A, it is considered a descendant of language A.
6. Additionally, linguists can create a word-level etymological dictionary, which considers factors like borrowing and semantic shifts.

This method is iterative, with updates made to cognate sets by identifying new sound correspondences and potentially modifying the selection of languages under examination.

2.6.2 Computational Methods

The comparative method in linguistics is a detailed and time-consuming process used to reconstruct ancestral languages and identify genetic relationships between them. It considers different aspects of language such as sounds, word forms, and sentence structures. This process can be challenging for linguists. Recently, computational methods have started to simplify this process. Researchers have tested various computational techniques, such as probabilistic models like Monte Carlo inference and Bayesian method to improve accuracy [23]. Also, Machine learning has shown promising results in enhancing how accurately and efficiently historical ancestor languages are reconstructed [9, 24].

The field of Natural Language Processing (NLP) has shown significant advancements from its early rule-based systems to the more sophisticated statistical and deep learning methods. The rapid development of NLP over the past few decades has revolutionized the field of machine translation and language understanding. The reasons for this advancement are LLMs and sophisticated machine translation architectures, particularly those based on deep neural networks.

The following sections will discuss the meaning of various computational methods and provide an overview of these technologies' evolution, their architectures, and their implications.

2.6.2.1 Natural Language Processing

The early developments in NLP and machine translation were predominantly driven by rule-based systems, where manually crafted rules and dictionaries played a central role in language processing. However, during the late 1980s and 1990s, there was a significant shift towards statistical methods. Models such as Hidden Markov Models and phrase-based translation systems began to gain prominence, leveraging probabilistic approaches to enhance the performance of language processing tasks [45, 46].

Hidden Markov Models are probabilistic models that decide the next state of a system based on the current state. For example, in NLP the proto-word can be suggested based on the cognate word character phonetic representation. This is done by using Markov Model by tracing the transition from phonemes in modern cognate words to their corresponding proto-phonemes in the parent language. The transition probabilities represent the likelihood of a phoneme in a modern language evolving from a proto-phoneme, while the emission probabilities capture the likelihood of observing the proto-phoneme given the modern cognate phoneme. By applying the Viterbi algorithm, the HMM identifies the most probable sequence of proto-phonemes that could have produced the observed phonemes across different cognate words, effectively reconstructing the original proto-word from its modern descendants.

The other rule-based Machine Learning (ML) method is Phrase-Based Translation. This is done by breaking down sentences into smaller phrases and translating them individually. These systems relied heavily on statistical methods to determine the most likely translation for each phrase, often resulting in disjointed and less fluent translations [45, 46].

2.6.2.2 Deep neural networks

The idea of a Deep Neural Network(DNN) comes from the concept of artificial neural networks, which are inspired by the structure and function of the human brain. A DNN extends this idea by incorporating multiple layers of interconnected neurons, or nodes, between the input and output layers. This layered architecture allows the network to learn complex patterns and representations in data by introducing non-linearity through activation functions and adjusting weights during training. By leveraging these deep structures, DNNs have become powerful tools for tasks such as image and speech recognition, and natural language processing [47, 48].

Neural networks are structured as interconnected nodes arranged in layers, as a complex web where each node communicates within its layers through specialized mathematical operations. These operations facilitate the network's ability to learn patterns, such as identifying faces in images or predicting words. Neural networks are esteemed for their capacity as powerful problem-solving tools, adept at uncovering and analyzing patterns, interpreting languages, and showcasing the efficacy of advanced algorithmic frameworks [47, 48, 49].

This section explores the primary neural network architectures used in proto-language reconstruction, focusing on their role in enhancing statistical methods or forming the foundation of neural machine translation approaches. These networks are mostly trained using supervised learning, where they learn patterns and relationships from large datasets of examples. In addition, In-context learning and large language models are also discussed.

Feed-forward Neural Networks

Feed-forward Neural Networks (FNNs), connect the inputs through hidden nodes to the outputs without loops. Basically, these networks can be classified into single and multi-layer perceptrons. Single-layer perceptron consists of a function that maps its input to an output value. The multi-layer perceptron consists of several fully connected layers in a directed graph. Each layer has several nodes, and each node is a neuron with a non-linear activation function [47, 50].

Convolutional Neural Networks (CNNs)

A popular type of FNNs are Convolutional Neural Networks (CNN)s, whose connectivity pattern between their neurons is inspired by the overlapping of the individual neurons of the animal cortex. A convolution operation is a mathematical way to describe this connectivity pattern. There are 3 basic properties of CNNs on top of FNNs which are: local connectivity (only adjacent neurons are connected), parameter sharing (replicated units share the same parameterization), and maxpooling units which is a form on subsampling [51].

Recurrent Neural Network

Recurrent Neural Networks (RNNs) are another class of neural networks. The main characteristic is that connections between units form a directed cycle, which generates an internal state with dynamic temporal behavior. FNNs typically rely on a fixed-size context window making the Markov assumption that a word only depends on n previous words. On the other hand, RNNs are able to use the internal memory to get rid of this Markov assumption and condition on all previous words, which is highly relevant in language modeling and MT. There are different types of RNNs and, in this manuscript, we focus on the most used in first neural MT systems [47, 52].

- **Long Short Term Memory (LSTM)** has the direct cycle structure with a different structure in the repeating cycle. The repeating cycle has three neural network gates (input, memory/forget, and output) which allow to discard or keep information solving the problem RNN faces on the vanishing gradient. Intuitively, the vanishing gradient problem may appear when using gradient-based and backpropagation methods. When training weights with these algorithms, these weights are updated using the gradient of the error function. At this point, and for RNNs, the chain rule is applied for the entire history of the sequence, and applying this many times may cause the gradients to tend to zero (specially, when using activation functions as tanh or sigmoid) [52, 53].

- **Gated RNN(GRU)**, An alternative to LSTMs is the Gated RNN, whose main difference is that instead of having three gates as LSTMs, GRUs have two gates (reset and update). GRUs have fewer parameters to train compared to LSTMs which may help in training faster and generalizing better with less data [52, 53]

- **Encoder-decoder:**Encoder-decoder architecture generalizes the idea of autoencoders allowing for having different input and output data. The encoder-decoder architecture aims at learning a representation (encoding) of input data, and decodes this representation while minimizing the amount of error for recovering the output data. The main purpose of the internal representation is a dimensionality reduction capable of extracting relevant features from the dataset [52].

In general sequence-to-sequence problems like machine translation and word prediction, inputs, and outputs have varying lengths that are unaligned. The standard approach to handling this sort of data is to design an encoder-decoder architecture consisting of two major components: an encoder that takes a variable-length sequence as input, and a decoder that acts as a conditional language model, taking in the encoded input and the leftwards context of the target sequence and predicting the subsequent token in the target sequence.

2.6.2.3 Large Language Models

Large language models have emerged as a transformative force in NLP, leveraging the power of deep neural networks to perform a wide range of language tasks with unprecedented accuracy and fluency.

In recent years Transformer architecture revolutionized Natural Language Processing. It utilizes self-attention mechanisms to process input sequences in parallel, addressing the limitations of RNNs in handling long-range dependencies. This architecture underpins most state-of-the-art LLMs, including BERT and GPT series.

Modern LLMs are characterized by their scalability, with models containing billions or even trillions of parameters. These models are pre-trained on massive datasets to learn general language representations, which are then fine-tuned for specific tasks. This pre-training-fine-tuning paradigm significantly enhances performance across various NLP applications.

In-Context Learning (ICL)

In-context learning (ICL) represents an innovative learning paradigm in which a language model observes a few examples and produces predictions for test inputs without the need for parameter updates [14]. In this approach, the model conditions a limited number of training examples to make direct predictions on new test data. Due to the constraints imposed by the model's maximum input length, it is common practice to randomly sample a small subset of examples from the full dataset for ICL. However, this method can lead to significant instability and reduced performance, as randomly selected examples may not be optimal. Therefore, selecting a concise set of examples that are both informative and representative of the entire dataset is crucial for effective in-context learning [14, 54].

There are two main areas of prompt design for In-Context learning.

- Demonstration organization: Focuses on selecting the best training examples for query(using random selection, Reinforcement learning, or embedding similarity) depending on the task.
- Demonstration Formatting: Involves the design of the prompt itself, including its language and structure.

Good instructions are key to getting the best results from in-context learning , whether using zero-shot, few-shot, or many-shot approaches. The way prompt is designed plays a big role in guiding the model's responses, and refining these prompts based on the model's output can make a huge difference in accuracy. It's also worth experimenting with different prompts to tap into GPT-3.5's and GPT-4o full range of abilities from answering questions to creative writing. Integrating the model via API into your workflows allows to make the most of its capabilities without the need for time-consuming retraining or fine-tuning [14, 55, 56].

2.6.2.4 Machine Translation Architectures

Machine translation has benefited immensely from the advancements in deep learning and the development of large language models. Early machine translation systems employed phrase-based approaches, breaking down sentences into smaller phrases and translating them individually. These systems relied heavily on statistical methods to determine the most likely translation for each phrase, often resulting in disjointed and less fluent translations.

Neural Machine Translation (NMT) has increasingly become the foundation of modern translation systems. In this setup, the encoder plays a key role by processing the source language sentence or word and converting it into a continuous representation. From there, the decoder takes over by creating the target output based on that representation. To further enhance the encoder-decoder model attention mechanism was introduced. This allows the decoder to dynamically focus on different parts of the source sentence to improve handling longer sentences and capturing context more effectively. This method not only allows for seamless end-to-end training but also produces more natural and fluent translations compared to older phrase-based techniques.

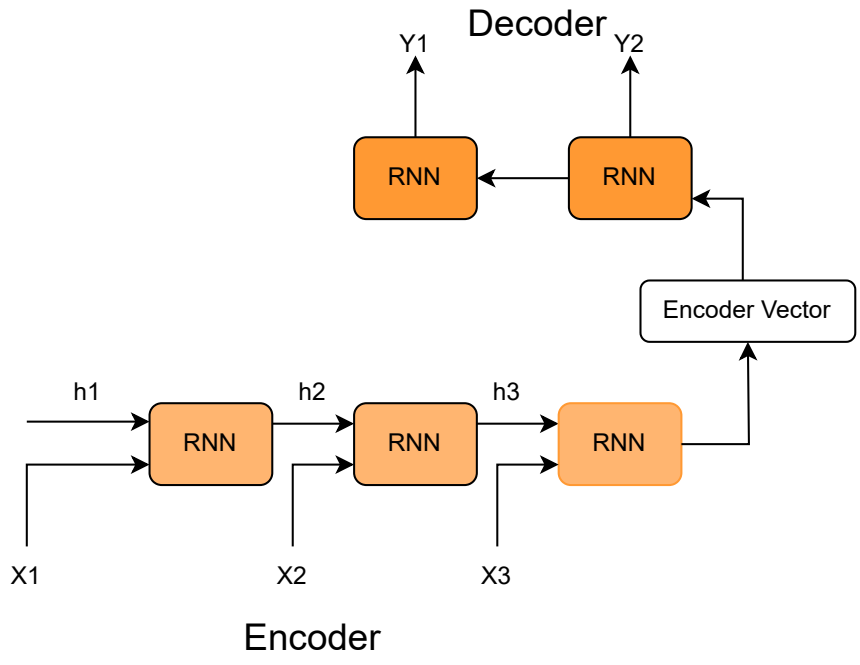

Figure 2.2: RNN based Encode decode

Transformer-Based Neural Machine Translation (NMT) has revolutionized the field of Neural Machine Translation. The introduction of the Transformer model represented a breakthrough by utilizing self-attention mechanisms and positional encoding. This approach allows the model to process entire sentences simultaneously, significantly enhancing both efficiency and translation accuracy. Transformer-based systems raise the standard for what's achievable in NMT.

2.7 Summary

Proto-word reconstruction focuses on recreating ancient or unrecorded words using cognates from current daughter languages. Despite challenges related to limited datasets, this task has gained popularity due to its potential in linguistic research. It involves multiple steps such as cognate detection, where linguists identify words in different languages that share a common root. The International Phonetic Alphabet (IPA) plays a crucial role in tracking phonetic changes and understanding sound evolution. Comparative and computational methods further enhance this process by systematically analyzing languages and applying algorithms to large datasets. Despite being more complicated than many other linguistic tasks proto-word reconstruction is advancing rapidly.

Chapter 3

Literature Review

Historical ancestor language reconstruction is a major research field in linguistics that has gained increasing attention in recent years. Researchers have applied various computational and comparative methods to reconstruct proto-words from which modern language words derive. This chapter reviews recent research papers that implemented historical ancestor language reconstruction using computational methods. In addition, different research papers implemented in-context learning using large language models for text classification, translation, and code generation are discussed. Cognate identification methods are also discussed.

3.1 Cognate Set Identification

Traditionally, identifying cognates requires meticulous manual comparisons of lexicons across various concepts, demanding significant linguistic expertise. Early cognate identification approaches rely mostly on phonetic similarities and sound correspondence. With the growth of large-scale linguistic datasets and computational power, computational approaches for cognate detection have become popular: methods that rely on edit distance [57], clustering [58], and expectation-maximization techniques [59] have become popular. The Lingpy toolkit [60] has continued to be a vital resource by offering methods for automated cognate detection through sound class-based alignment and phonetic distance measures, enabling combining traditional linguistics insights with computational power. By integrating deep learning and advanced computational models with traditional linguistic methods [61] introduced transformer-based models that show better performance in capturing phonetic and contextual similarities across languages, outperforming traditional alignment techniques. Yet, challenges remain for ancient and low-resource languages with little to no collected data and linguistic experts.

3.2 Historical Language Reconstruction

One of the influential and eye-opening papers in protolanguage word form reconstruction is the work of L. Bouckaert et al. [62]. The paper presents an unsupervised approach to reconstructing ancient protolanguages from their modern descendant languages. Their proposed method addresses the improved performance by incorporating faithfulness and markedness features. It also introduces a novel Markov Chain Monte Carlo inference procedure called Ancestry Resampling that addresses the mixing problems that can arise in large phylogenetic trees. [23] Used Bayesian phylogenetic methods to reconstruct the language tree for the Austronesian family, which includes over 1,200 languages spoken in the Pacific region.

3.2.1 Deep learning methods

In recent years, rule-based machine learning methods have been replaced by deep neural networks. Dekker [47] explores the application of machine learning in historical linguistics, focusing on pairwise word prediction and modeling sound change to reconstruct linguistic ancestry. It introduces a method where a machine learning model is trained on phonetic word pairs from two languages to learn sound correspondences and predict word forms accordingly, utilizing both a sophisticated RNN encoder-decoder and a simpler structured perceptron. The outcomes including prediction distances and model parameters are applied to significant tasks such as phylogenetic tree reconstruction, sound correspondence identification, and cognate detection. To enhance the model's effectiveness the authors propose a "cognacy prior" loss function to prioritize learning from cognate pairs. Experimental results using data from Slavic and Germanic languages show that model performance is comparable to baseline techniques, though increased complexity does not guarantee better results. The paper marks an innovative use of machine learning in addressing complex historical linguistics challenges.

Another paper by Ciobanu et al. [63] introduces a novel technique for proto-word reconstruction from cognate sets in various Romance languages, aiming to develop a tool for historical linguistics that facilitates the analysis of produced proto-words by experts. The methodology employs conditional random fields leveraging character n-grams as features to infer Latin ancestor forms from modern word pairs. It was done by utilizing a sequence labeling approach that requires only (word, proto-word) pairs as input. Two alignment methods profile hidden Markov models and the Needleman-Wunsch global alignment algorithm were evaluated showing the latter performing slightly better results. To enhance accuracy ensemble methods were employed, combining outputs from classifiers trained in different modern languages. They were able to demonstrate significant performance improvements, achieving an average edit distance of 1.07 and listing the correct proto-word among the top five predictions 70% of the time. The methods require less input data than previous approaches and can accommodate incomplete data, making them especially useful for low-resourced languages.

Meloni et al. [9, 64] Proposed a novel approach to reconstructing proto-languages using deep learning techniques. The authors argue that traditional linguistic methods have limitations in their ability to accurately reconstruct ancient languages, especially for languages that have been extinct for thousands of years. The paper details a neural network model that has been developed to reconstruct the proto-forms of ancient languages by processing large amounts of linguistic data. They implemented a model traditionally used in machine translation for the reconstruction of Latin words using its forms in French, Portuguese, Italian, Romanian, and Spanish. The architecture consists of an encoder and a decoder both of which are recurrent neural networks (specifically,GRU). Additionally, the decoder uses the mechanism of attention. The input for this algorithm is character embedding concatenated with the vector representations of a language.

Nitschke et al. [65] proposed a new approach that uses modern related languages or sisters to reconstruct the vocabulary of a target language. The paper demonstrates that this can be achieved using a relatively small dataset of parallel cognates from various sister languages by employing a neural machine translation (NMT) architecture with a standard encoder-decoder setup. This initiative seeks to leverage machine learning tools to support underserved language communities in reclaiming, preserving, or reconstructing their languages.

Andre He, Nicholas Tomlin, and Dan Klein [66] propose a novel method for reconstructing the word forms of a protolanguage. The method is based on training a neural network to predict how words in modern languages might have evolved from their proto-language forms. The network is trained using unsupervised learning without relying on explicit information about the relationships between languages or the meanings of words. The authors evaluated their method on a dataset consisting of 31 languages from the Austronesian language family and showed that it is able to reconstruct protolanguage word forms with high accuracy compared to previous methods. They also demonstrate that the method can be used to identify sound changes and reconstruct sound correspondences between languages, which can shed light on historical relationships between languages. The approach presented in this paper requires a large amount of training data, which may not be available for all language families. This could limit the applicability of this approach for reconstructing proto-language word forms for some language families.

Kim et al. [10] discusses the challenges associated with reconstructing unattested proto-languages and the application of the Transformer architecture in achieving state-of-the-art performance in protoform reconstruction. The authors focus on systematic sound changes, data scarcity for under-documented languages, and the expansion of the Chinese dataset to include 804 cognate sets across 39 modern varieties and Middle Chinese. The paper introduces the task of protoform reconstruction and highlights the inherent data constraints for reconstructing unattested proto-languages. Previous attempts at applying machine learning to this task included supervised and unsupervised approaches, but data scarcity remained a limiting factor. They expanded the Chinese dataset to include 804 cognate sets across 39 modern varieties and Middle Chinese. The study includes a comparative analysis of the Transformer model with various baseline models, demonstrating its superior performance in protoform reconstruction across multiple metrics and datasets. The paper also discusses the limitations and challenges of the study, including the reliance on existing reconstructions, the need for substantial amounts of data, and the potential variations in performance between models. Additionally, the authors address the limitations of their Chinese dataset, particularly in terms of the source of the Middle Chinese protoforms.

The most recent paper by Cui et al. [67]investigates advanced methods for proto-form reconstruction in linguistics, particularly addressing challenges posed by incomplete cognate sets in datasets like WikiHan. The authors implement three main approaches: first, data augmentation techniques that predict missing daughter forms based on existing entries and proto-forms to enhance training stability and model performance; second, reflex prediction using a CNN model originally designed for image inpainting by treats daughter form prediction like pixel recovery, demonstrating that effective reflex prediction can improve model robustness and yield slight performance gains, although it doesn't surpass prior benchmarks; and third, a character-level transduction method employing a transformer model that predicts daughter forms using the proto-form and target language as inputs, enhancing contextual understanding through additional feature embeddings. While integrating Variational Autoencoder VAR structures improved transformer performance, the study suggests further exploration of RNN models and optimization of daughter form utilization to better reconstruct proto-forms.

3.2.2 In-context Learning

In-context learning ICL has significantly advanced machine learning in natural language processing NLP. The in-context learning approach, introduced by Brown et al. [14] has deepened the understanding of what large language models can do. It showed that these models can handle complex tasks by using just a few examples (few-shot) or sometimes even none at all (zero-shot) simply by formatting the input correctly. This skill highlights how well LLMs can generalize across various tasks like machine translation and question answering without needing any updates to their underlying models.

For instance, [68, 69] showed that incorporating contextual examples improved text classification accuracy. Additionally, Li et al. [70] and Y. Chen et al. [71] demonstrated that ICL refined answers in question answering tasks, and A. Patel et al. [15] highlighted its potential in code generation.

Effective prompt engineering has emerged as a key methodological approach, optimizing model responses. To use incontext learning one major issue is that LLMs are extremely sensitive to the quality of the specific examples provided in-context [55, 56]. If these examples don't accurately represent the task at hand, the model may struggle to learn effectively. As ICL continues to evolve, addressing these challenges will be crucial for its broader application and impact in the field. Challenges such as dependency on context quality and scalability across diverse tasks remain.

3.3 Summary

This chapter covers a comprehensive literature review of the existing works on computational methods used to perform cognate identification, proto-word reconstruction, In-context learning, and their proposed solutions.

Chapter 4

Methodology

This chapter discusses the methodology implemented for cognate detection and proto-language reconstruction of Ethio-Semitic languages. It begins with a detailed overview of the dataset, covering its sources, characteristics, preprocessing, and cognate identification. It also explains the feature extraction methods used to identify relevant linguistic features. The role of Large Language Models and machine translation architecture in predicting sound changes is highlighted. This chapter also focuses on the reconstruction units, particularly phonemes, and outlines the evaluation metrics used to assess system accuracy. By addressing these components, this chapter provides a detailed summary of the procedures and methods employed in creating cognate datasets and sequence-to-sequence encode-decode LSTM-based model for the proto-word reconstruction of Ethio-Semitic languages.

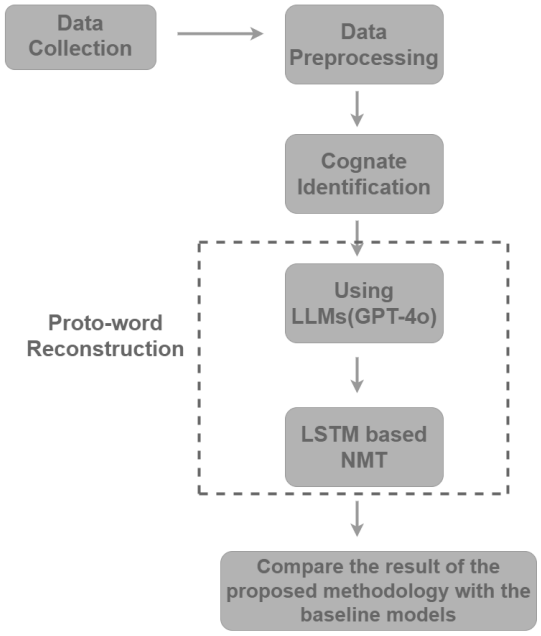

Figure 4.1: The proposed model architecture for Reconstruction of Proto-EthioSemitic.

4.1 Dataset

Data from various linguistic levels can be utilized to study language change, including lexical, phonetic, and syntactic data. Word forms (lexical or phonetic) are particularly suitable for prediction tasks. Allowing the prediction algorithm to generalize the relationships between phonemes by preparing a cognate dataset. Words in phonetic representation form are used because they can closely reflect the actual language used by speakers [47].

4.1.1 Data collection

As discussed in Section 2.3, data was collected in three Ethio-Semetic languages: Amharic, Tigrinya, and Ge'ez. Data was collected in two stages: First, for the golden data set words from the Swadesh list [72, 73] concept were translated into the three languages of interest. The Swadesh list is a compilation of tentatively universal concepts for lexicostatistics. The list was put together by Morris Swadesh on the basis of his intuition [73]. This list contains 100 terms. The words in the Swadesh lists were chosen for their universal, culturally independent availability in as many languages as possible, regardless of their "stability".

thou	we	this	that	who	what	not	all	many
louse	tree	seed	leaf	root	bark	skin	flesh	blood
mouth	tooth	tongue	fingernail	foot	knee	hand	belly	neck
kill	swim	fly	walk	come	lie	sit	stand	give
fire	ash	burn	path	mountain	red	green	yellow	white

Table 4.1: Swadesh list examples(45 out of 200)

Secondly, for the main data set words with the same meaning across all languages of interest were collected from dictionaries. Data was collected from lsanat sem [74]. A total of 14,100 words were collected across the three languages. The final dataset includes words with different suffixes and prefixes of a single morpheme across the three languages; allowing us to capture sound change in different morphological units. Words included in the final dataset are selected based on:

1. Having the same meaning
2. Not being borrowed/loan words
3. Being currently used by speakers of each of the languages

4. Having meaning that can be represented by a single word

4.1.2 Data Preprocessing

Pre-processing is a vital step in pattern detection and prediction researchs because it ensures that the results are accurate and reliable [75, 47]. This stage involves transforming raw data into a format that is easy to analyze for the proposed methodology. The main tasks in data pre-processing include cleaning the data, converting it from one format to another, and tokenization.

4.1.2.1 Data cleaning

Data cleaning is a crucial process that involves the removal of incomplete data, the correction of formats, and the handling of missing values. To perform an effective pattern detection the dataset needs to be accurate, complete, and consistent. The primary objective of data cleaning is to eliminate all possible errors and inconsistencies that could result in biased results from the data [76]. Data cleaning was performed as follows:

- Removing special characters in the text, such as punctuation marks or non-alphanumeric symbols. This includes characters unique to the language's script, such as ፡ (Ethiopic word space) or ። (Ethiopic paragraph separator). Removing these characters helps standardize the text and prevents potential issues during further processing steps.
- Removing instances of redundant or duplicated words to streamline the text and reduce redundancy. For example, if the same word appears m ultiple times consecutively or if unnecessary symbols like repeated punctuation marks are present, they are removed. This process ensures that each word and symbol in the text serves a meaningful purpose and contributes to the overall clarity of the content.
- Due to data formatting issues or errors, multiple words might be incorrectly combined into a single cell or unit of text. By splitting these instances, each word is properly segmented and treated as an individual unit, which facilitates accurate analysis and processing downstream.

- Words in the dataset that are contextually ambiguous or have multiple meanings are manually corrected to ensure accurate representation. This process may involve linguistic expertise or context-based analysis to disambiguate the intended meaning of each word. By manually correcting such words, semantic coherence is maintained and ambiguity is reduced, which is crucial for reliable analysis and interpretation.

These cleaning steps aim to enhance the quality, consistency, and interpretability of the dataset.

4.1.2.2 Feature Extraction

Feature extraction is a technique in machine learning and data analysis that focuses on identifying and extracting significant features from raw data. It goes a step further by transforming the data into a format that is more suitable for analysis. This process can involve dimension reduction, identifying meaningful patterns, or converting the raw data into a different format to uncover essential characteristics. These features are subsequently used to construct a more informative dataset [77, 75, 47].

After cleaning the data, the individual words were converted to their IPA format, so that it can be easier to extraction more detailed sounds representation. Converting words to their IPA is crucial for the linguistic task of proto-language reconstruction. The words are converted to their phonetic representation using Epitran Python library[1]. For both Amharic and Tigrinya, the more phonemic IPA code as mentioned in Table 2.1 is used to capture a wider range of phonetic features. For Ge'ez, the Tigrinya more phonetic code is used since it has more unique phonetic representations for the characters in the Ge'ez script that all the languages use[2].

4.1.2.3 Tokenization

Tokenization is the process of breaking down text into smaller units called tokens. Tokens can be words, sub-words, characters, or other meaningful elements [78, 79]. The IPA representation for the words has to tokenize in character lever which splits the text into individual characters IPA phonetics so that it can be easier to identify the sound changes of the three daughter languages words for the same concept.

[1] https://github.com/dmort27/epitran

[2] For example, in the Amharic language there are 4 different letters for the glottal stop sound "ʔa" while in Tigrinya those four letters each have a distinct sound: "ʔa", "ʔɐ", "ʕa", and "ʕɐ"

Swadesh Word	Ge'ez	Tigrinya	Amharic
All	ኩሉ kʷilu	ኩሉ kulu	ሁሉ hulu
Egg	እንቁቅሆ ʔəniqoqiho	እንቁላሊሕ ʔiniqulaliħ	እንቁላል inik'ulal
Eye	ዐይን ʕəjin	ዓይኒ ʕajini	ዓይን ajin
Name	ስም sim	ሽም ʃim	ስም sim

Figure 4.2: Samples from the Swadesh 100-word list in the three languages of study(Gold data set).

	Amharic	Tigrinya	Ge'ez	Total
Golden data (From Swadesh List)	100	100	100	300
Main dataset (From Dictionary)	14,100	14,100	14,100	42,300

Table 4.2: The number of words collected for each language using the two ways.

4.2 Cognate Identification

For cognate identification, a multi-faceted approach integrating both computational from section 2.5.2.1 and linguistic techniques is employed. A compiled dataset holding 100 Swadesh list concepts with their meaning across Amharic, Tigriniya, and Ge'ez was given to linguists. Through phonological and morphological criteria which is the traditional comparative method, the linguists identified 74 out of the 100 concepts as cognates.

Then three different LingPy cognate detection techniques which are SCA, Turchin, and Lexicon were used on the data that were verified by the linguists (henceforth referred to as "Gold Dataset") and results were used to determine which method was best suited for our language family. After evaluating the effectiveness of each technique, the most suitable methods were selected for further analysis which are SCA and LEXSTATID. These chosen methods were then applied to the remaining data. Our approach integrated computational tools and manual verification, with linguistic experts reviewing the results to account for historical sound changes and semantic shifts. This robust methodology facilitated a reliable identification of cognate sets, enhancing our understanding of the historical and evolutionary relationships among the languages studied.

Through the above process, two datasets are built:

1. Gold Dataset (Identified by linguists)
2. Main Dataset (Identified by lingpy method and verified by linguists)

Step 1 **Sematically Related Words**	**Step 2** **Cognate Identification**	**Step 3** **Proto-Form Reconstruction**
Ge'ez	**IPA**	**IPA**
Swadesh List: 100 concepts 300 words.	Linguist: 74 cognates	Linguist: 74 proto-form
Dictionary: 14,100 terms. 54,300 words	Automatic:1847 cognates	Syntetic: 1847 proto-form

Table 4.3: Steps taken to create the dataset and the number of words, cognates, and proto-forms created at each step.

4.3 Proto-EthioSemitic Reconstruction

To our knowledge, no prior proto-language reconstructions were done for the Ethio-Semitic language family. Hence, reconstructing its proto-form from the existing daughter languages is novel research. First, in-context learning using large language models is examined on the reconstruction process to generate synthetic data set. Then the existing methodologies for other language models are implemented on the generated synthetic data set. Finally, Using the proposed LSTM-based encode-decode model, the proto word of Ethiopian-Semitic languages are reconstructed.

4.3.1 In-context learning

LLMs are models usually employing the Transformer architecture that are trained to understand, summarize, generate, and predict text-based content using deep learning techniques and massive datasets. These models are specifically engineered to process and generate 'human-like' text, making them powerful tools for many tasks in natural language processing. But since (1) LLMs are too large to train from scratch and (2) already contain large amounts of knowledge, methods like In-Context Learning are used to utilize LLMs for different tasks with complex structure and scarce data set [80, 54, 81, 14].

In-context learning (ICL) is an approach where a language model makes predictions using just a few examples, without needing to train the model [14]. This method is particularly useful when labeled data is limited, as it can handle tasks with minimal input. However, randomly selecting examples due to input length constraints can lead to less stable and less effective results [54, 82]. To maximize the effectiveness of ICL, it's crucial to carefully select examples that are both informative and representative of the broader dataset. In ICL (In-Context Learning), the average preferred number of examples is around eight. However, if the examples are positive and the task is complex, increasing the number of examples for better presentation of the dataset can improve the model's performance [83, 84].

To implement in-context learning, two specifically constructed data sets were utilized(see Section 4.2). The Gold Dataset served as the benchmark for testing the model's in-context character capabilities and assessing its performance in language reconstruction. After validating the model's effectiveness, it was then applied to reconstruct the proto-forms of the Main Dataset, which excluded ancestral words.

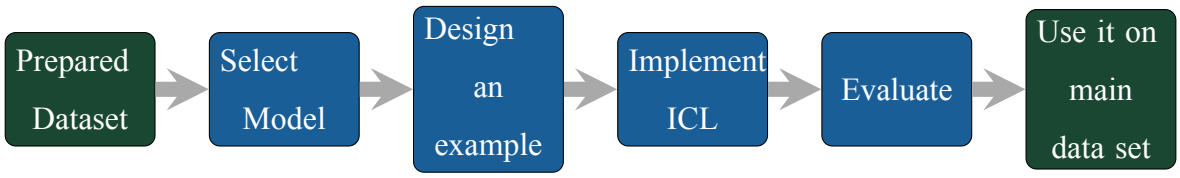

Figure 4.3: Flow Diagram of the In-Context Learning Process

The initial step in this process is selecting a suitable large language model (LLM). W. Chen et al. [85] and Brown et al.[14] showed that the size of parameters models trained on plays a crucial role in determining its performance for in-context learning tasks. Another essential factor to consider is the model's availability. GPT-4o, with an estimated parameter range in the hundreds of billions to trillions(see Table 4.4) stands out for its substantial parameter size, even though it is a closed-source model.

Model	Size (Parameters)
GPT-3.5	175 billion [14]
GPT-4	Unknown, estimated in trillions [86]
GPT-4o	Unknown, estimated in trillions and is even more efficient than GPT-4 [86]
LLaMA	7 billion to 70 billion [87]
BERT	110 million (Base), 340 million (Large) [88]
BLOOM [89]	176 billion
Aya [90]	13 billion

Table 4.4: Sizes of Various Large Language Models (LLMs)

The second criterion for selecting the most suitable language model for this task involved evaluating the performance of three LLMs on proto-word reconstruction using the golden dataset.

The next step is to design an example, a process known as prompt design or demonstration design. The number of demonstration examples used can significantly impact a model's performance, depending on the complexity of the task. For simpler tasks, adding more demonstrations often does not lead to noticeable improvements. On average, using around eight or under demonstrations tends to yield better results [91]. However, for more complex tasks such as language reconstruction, the model needs to understand sound pattern changes across cognates. In such cases, the number of demonstrations should be carefully selected based on how well they highlight these phonological patterns. In this step, 11 sequences were randomly chosen, with each sequence comprising 11 labeled examples, each sequence consisting of examples, $((x_1, y_1)), (x_2, y_2)), \ldots, (x_m, y_m))$, where x_m is a sequence of lang1, lang2, and lang3, and y_m is the corresponding proto-form. lang1, lang2, and lang3 represent Ge'ez, Tigrinya, and Amharic correspondingly. The selection of these examples was done randomly, guided by the linguists' judgment of which examples best represent the contextual information of the data in the data set.

The demonstration examples are prepared as follows:

- Sampled 11 sequences, each containing 'lang1', 'lang2', 'lang3', and their corresponding 'proto' forms.

- Constructed input-output pairs for each sequence as $(x_1, f(x_1)), (x_2, f(x_2)), \ldots, (x_m, f(x_m))$, where x_i represents a sequence in 'lang1', 'lang2', and 'lang3', and $f(x_i)$ represents the corresponding 'proto' form.

For in-context learning, each sequence is formatted into a prompt that the LLM can understand. The prompts were structured as follows:

Examples = [

"prompt": "Language A: `k$^{w}$illu', Language B: `kulu', Language C: `hullu', Proto-word: ``kullu"",

"prompt": "Language A: `ʔəb', Language B: `ʔabo', Language C: `abat', Proto-word: ``ʔab"",

...]

Each example within the prompt provides the LLM with multiple input sequences ('lang1', 'lang2', 'lang3') along with the corresponding 'proto' output. The first 11 examples are used as the context for the LLM using prompt design. These examples demonstrate how the input sequences map to the 'proto' sequences.

Following the prompt design, the next step involves applying ICL. This is achieved by feeding the prepared prompt examples into the model to assess GPT-4o's performance in reconstructing proto-words. The evaluation focuses on how well GPT-4o reconstructs proto-forms for the remaining cognates in the golden dataset. In addition to GPT-4o we experimented on two generative models which are mT5 and AfriTeVa. And GPT-4o performed well compared to those models.

Based on the model's performance in testing, GPT-4o is then employed to reconstruct the ancestor language of Ethio-Semitic words main data set which is the data set that does not have a proto-form.

4.3.2 LSTM-based encode-decode model

Neural Machine Translation (NMT) models generally feature an encoder-decoder architecture. The encoder processes the input words, which can vary in length, and converts them into a fixed-length representation. This representation captures the essence of the input word characters. The decoder then takes this representation and generates the predicted word in the target proto-language. This approach allows for more accurate and fluent translations by leveraging deep learning techniques to understand and produce language [92, 93].

Language reconstruction is the process of inferring the features of an ancestral language (a proto-language) from its descendant languages. This task requires models that can effectively capture the relationships between different languages and sequences of linguistic features.

In this research paper, we propose LSTM-based encode-decode model for proto-language reconstruction. LSTM networks are a type of recurrent neural network (RNN) designed to handle the problem of vanishing and exploding gradients, which are common issues in traditional RNNs [94]. The model consists of two main parts: an encoder and a decoder. The encoder processes the input sequence (e.g., a word or sequence of sounds in a daughter language) and generates a series of hidden states that encapsulate the linguistic information from the input. The decoder then takes these hidden states and reconstructs the proto-language sequence, generating the ancestral word or sound sequence one element at a time [92, 93].

This research aims is to predict and reconstruct proto-forms (ancient or reconstructed language forms) from their cognates across the three Ethio-semitic languages. This model leverages a sequence-to-sequence (seq2seq) architecture with attention mechanisms, embedding layers, and Multilayer Perceptron (MLP) networks to capture the relationships between languages and accurately predict the proto-form.

- **Encoder-Decoder Framework:** The core of the model is a sequence-to-sequence architecture that includes an encoder and a decoder, both implemented using LSTM networks. The encoder processes the input sequence (e.g., a word from a descendant language) and converts it into a series of hidden states. The decoder then uses these hidden states to generate the output sequence, which could be the reconstructed word in the protolanguage.

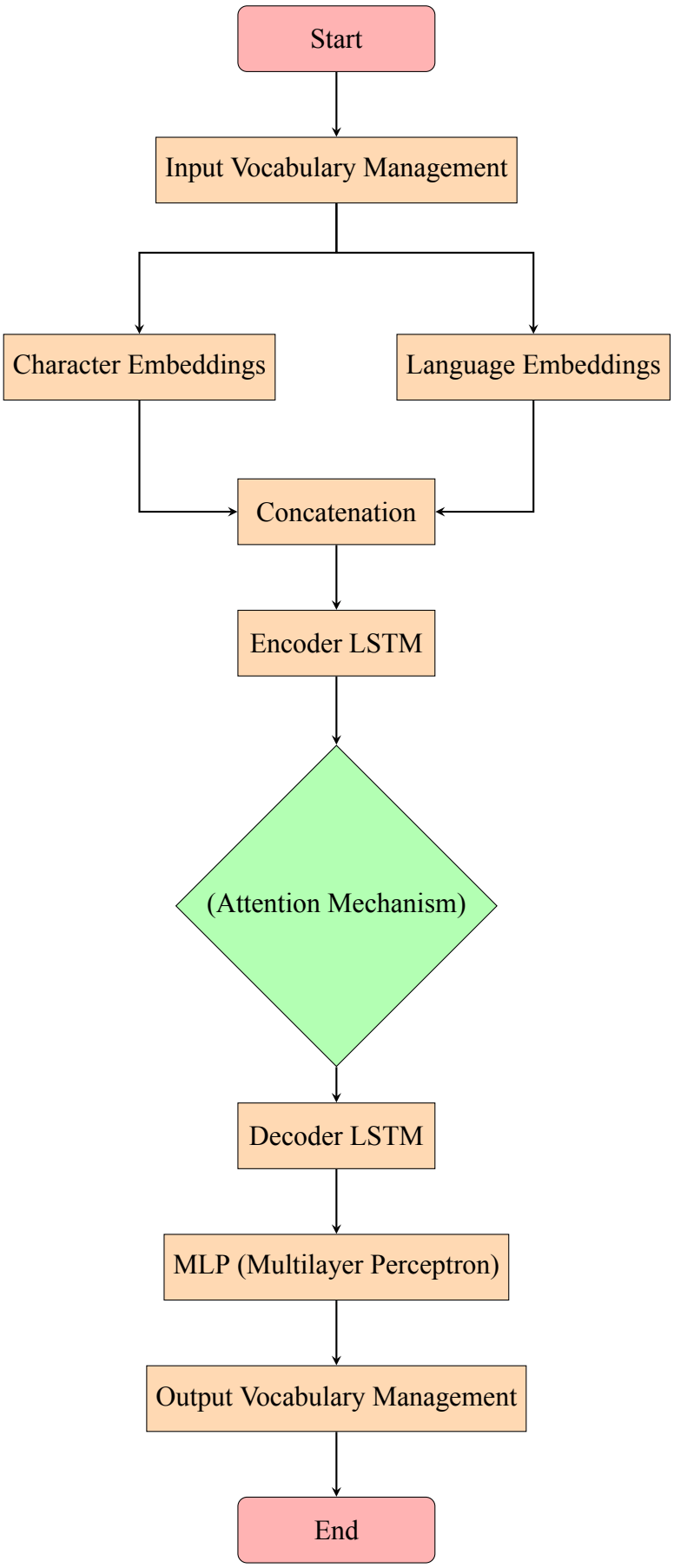

Figure 4.4: The proposed LSTM-based encode-decode architecture flow diagram

- **Embedding Layer:** The model uses a sophisticated embedding system to represent characters and languages in a dense vector space. These embeddings provide the model with a rich, context-aware representation of the input data, which is crucial for capturing the subtle differences between languages and predicting the correct output sequence.

- **Attention Mechanism:** To improve the model's ability to generate accurate outputs, an attention mechanism is integrated into the decoder. This mechanism allows the model to focus on the most relevant parts of the input sequence during each step of the decoding process, enhancing its ability to handle long and complex sequences.

- **Multilayer Perceptron (MLP):** After the decoder generates hidden states, they are passed through a Multilayer Perceptron to produce a probability distribution over possible output characters. The MLP refines the decoder's output, helping the model make accurate predictions at each step of the sequence.

- **Vocabulary Management:** A robust vocabulary management system is implemented to handle the various characters across different languages, including unknown tokens. This ensures that the model can operate effectively even with incomplete or noisy data, which is common in language reconstruction tasks.

Both the encoder and decoder are LSTM network. The forms of the words in the daughter languages are read by the encoder, and a contextualized representation of each character is output. At each decoding step, the encoder's representations are attended to by the decoder via a dot-product attention mechanism. The output of the attention mechanism is then fed into an MLP, which generates the next proto-EthioSemitic character.

Instead, the character embedding table is shared across all languages, including Proto-EthiSemitic. To each character vector, a language-embedding vector is concatenated.

4.3.3 Train-Test Split

To train the model built the data has to split to train, validate, and test. In this method, a major portion of the overall dataset is selected for training, and the remaining data is saved for validating and testing the models we built. The data set is split into 80/10/10, as training, validating, and testing respectively.

Language family	Total No	Training	Validating	Testing
Ethio-Semitic	1847	1477	185	185
Latin	5419	4335	542	542
Chinese	804	644	80	80

Table 4.5: Train-Validate-Test data split for the model.

The Meloni et al. [64] dataset, consisting of 8,799 cognate sets of Romanian, French, Italian, Spanish, and Portuguese words along with their corresponding Latin forms. However, only 5,419 cognate sets from the phonetic version (Rom-phon) were accessible. Additionally, the Chinese dataset includes 804 cognate sets from 39 modern Sinitic languages, along with their reconstructed Middle Chinese counterparts are used [10].

4.3.4 Experimental Parameters Setup

ParameterS	LSTM-based NMT
batch_size	1
beta1	0.9
beta2	0.999
dropout	0.27699
embedding_size	80
epochs	100
eps	1e-08
feedforward_dim	100
lr	0.000822
num_layers	1

Table 4.6: Parameters used for the proposed model.

Parameters and their setups are carefully chosen for better performance. In addition, different hyperparameters have been used to fine-tune the Deep Learning models. The setup used in the system is given in Table 4.6 above. Dropout is used to prevent overfitting.

4.4 Experimentation Setup

While doing the experiments we used the following:

Category	**Details**
Operating System	Ubuntu 22.04.2 LTS Server
Software	• Python 3.9.16 • pytorch 2.12.0 • lingpy 2.10
Hardware	• Dell Precision 7920 Tower server • Intel(R) Xeon(R) Gold 6230R CPU • 64GB RAM • NVIDIA RTX A4000 GPU

Table 4.7: Experimental Setup

As shown in Table 4.7 experiments were conducted on a system running Ubuntu 22.04.2 LTS Server as the operating system. The software stack included Python 3.9.16 for programming and scripting, PyTorch 2.12.0 for deep learning tasks, and LingPy 2.10 for linguistic analysis. The hardware configuration consisted of a Dell Precision 7920 Tower server equipped with an Intel(R) Xeon(R) Gold 6230R CPU, 64GB of RAM, and an NVIDIA RTX A4000 GPU, providing the computational resources necessary for the experiments.

4.5 Evaluation metric

Several evaluation metrics are used to assess the proposed methodology's performance. Evaluation metrics are quantitative tools used to evaluate a machine-learning model's effectiveness. These metrics offer valuable insights into the model's performance, allowing for a clear comparison between different models or algorithms [95]. In this research, we used the following evaluation metrics:

1. **Edit distance**: Edit distance quantifies how dissimilar two strings (e.g., words) are to one another by counting the minimum number of operations required to transform one string into the other. The allowable operations are addition (i.e. adding a character), deletion (i.e. removing a character), and substitution (i.e. replacing one character with another) [57]. Normalized edit distance (edit distance normalized by the length of the target) is employed to measure the similarity between the predicted and target protoforms.

2. **Accuracy:** Refers to the percentage of protoforms that are reconstructed without any mistakes. It is a measure of how often the reconstructed protoforms perfectly match the expected or true protoforms. It measures the overall correctness of predictions made by the model [24].

$$\text{Accuracy} = \left(\frac{\text{Number of correct reconstructions}}{\text{Total number of protoforms}} \right) \times 100$$

3. **Recall:** Measures the proportion of actual positives correctly identified by the model. In other words, it measures how well the system can identify cognates without missing. A high recall rate is important to minimize the number of false negatives. It is calculated as the ratio of the number of correctly identified cognates to the total number of cognate words [95].

$$\text{Recall} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalseNegative}}$$

4. **F1-score:** F1 Score is the harmonic mean of Precision and Recall, providing a balanced measure. A high F1 score indicates a balanced trade-off between precision and recall. [95].

$$\text{F1-score} = \frac{2 * (\text{precision} * \text{recall})}{\text{precision} + \text{recall}}$$

5. **Precision:** Measures the accuracy of positive predictions out of all instances classified as cognates. It reflects the ability of the system to correctly identify cognates. It is calculated as the ratio of the number of correctly identified cognates to the total number of words identified as cognates [95].

$$\text{Precision} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalsePositive}}$$

6. **Margin of error:** Is a measurement of the amount of error that arises from random sampling. It is considered important because it reflects the reliability of the results. By understanding the margin of error the level of uncertainty in the findings can be recognized, allowing for a more informed interpretation of the data.

$$\text{Margin of Error (ME)} = Z \times \frac{\sigma}{\sqrt{n}}$$

Where:

- **TruePositive:** is the number of correctly identified cogantes.
- **TrueNegative:** is the number of correctly identified not cognate words.
- **FalsePositive:** is the number of words incorrectly identified as cognates.
- **FalseNegative:** is the number of words incorrectly identified as not cognates.
- **Z** is the Z-score corresponding to your desired confidence level (e.g., 1.96 for a 95% confidence level).
- **σ** is the standard deviation of the prediction errors (or sample standard deviation if the population standard deviation is unknown).
- **n** is the number of predictions or sample size.

4.6 Summary

In this chapter, the working principles and methodologies applied in three key stages have been discussed: cognate identification, synthetic data generation, and the proposed approach for proto-word reconstruction. The process of identifying cognates from the existing dataset has been outlined. The methodology for generating synthetic data based on the identified cognates has also been described to expand and enhance the dataset. Lastly, the proposed methodology for reconstructing proto-words has been detailed. The results obtained from the experiments will be presented in the next section.

Chapter 5

Result and Discussion

The findings from the investigation into Ethiopian Semitic language reconstruction are presented in this section. In-context learning with GPT-4o and the LSTM-based encode-decode model were utilized. The model was evaluated using several performance metrics and compared to baseline models.

5.1 Cognate Identification Method Selection

As discussed in Section 4.2, cognate identification is the first step in this study. Using the methods described in Section 2.5.2.1, Table 5.3 presents the results for the three automatic cognate set discovery methods used. First, the Gold Dataset was used to select the best-performing method from the three automated cognate discovery methods. Then, the two best-performing methods were used to identify cognates for the Main Dataset.

	TURCHINID	LEXSTATID	SCAID
Precision	1.00	0.96	0.85
Recall	0.53	0.60	0.90
F1 Score	0.69	0.74	0.87
Accuracy	0.66	0.69	0.81

Table 5.1: Result

Based on the evaluation of the three identification methods, SCAID and LEXSTATID emerge as the preferred choices for different aspects of cognate identification. SCAID is chosen for its overall reliability, as evidenced by its highest accuracy of 81.1% and an impressive F1 score of 0.873 in addition to 90 % recall. This indicates that SCAID is highly effective in balancing the identification of true cognates (high recall) while minimizing incorrect predictions (high precision). Its ability to accurately predict both positive and negative instances makes it a suitable option for comprehensive cognate identification.

LEXSTATID, while slightly less accurate with a 68.9% accuracy is chosen for its moderate yet consistent performance. It serves as a valuable secondary method, offering a different perspective that complements SCAID's strengths. The choice of SCAID for its overall reliability and LEXSTATID for its solid performance ensures a well-rounded approach to cognate identification, combining the strengths of both methods for a more accurate and reliable outcome. Based on this performance on the Gold Dataset the two methods SCAID and LEXSTATID were applied on the Main Dataset which includes 14,100 entries. This process resulted in 1847 cognate words identified out of the 14,100 words triples.

Out of 1847 Linguists then verified a 196 subset of the automatically identified cognates, finding errors in just 2% of the samples. This low error rate indicates that, even though the automatic cognate identification is not perfect, it can correctly identify a large set of cognates. The final dataset from this step has 1847 cognates identified by automated methods and 74 cognates identified by linguistic experts.

5.2 Proto-word Reconstruction using GPT-4o

GPT-4o was choosen in addition to having the largest parameters trained on we additional experiment on other two generative models (mT5 and AfriTeVa) that are trained on Amharic language. We find that GPT-4o outperforms AfriTeVa and mT5. We hypothesize this could be because GPT-4o is trained on a wider range of language varieties inaddition to IPA and hence might more easily learn the patterns in IPA format from the small training data we provided. While both mT5 and AfriTeVa include Amharic in their pre-training, we are training and testing in IPA format hence the prior knowledge of the language may not be as relevant.

Model	**Linguist**
AfriTeVa-base	12.23
mT5-base	57.74
GPT-4o	**85.0**

Table 5.2: Model performance on linguist reconstructed test set.

From the Gold Dataset which has 74 words reconstructed by linguists, 11 were used as examples for In-Context learning, and the remaining 63 were used to test performance.

The proportion of correctly predicted instances out of the total number of instances using GPT-4o are able to achieve an accuracy of 85% with an error rate of 15%. Error Rate

	GPT-4o
Precision	0.84
Recall	0.85
F1 Score	0.84
Accuracy	0.85

Table 5.3: Result of LLM

is the proportion of incorrectly predicted instances out of the total number of instances evaluated. And it is calculated as:

$$\text{Error Rate (\%)} = 100\% - 85\% = 15\% \tag{5.1}$$

A 15% error rate implies that there will be instances where the model makes incorrect predictions. In addition to the error rate to strengthen the discovered result, the error margin is used as an evaluation metric.

Using this methodology, a synthetic proto-word was constructed for the main cognate dataset, which consists of 1,847 words for each of the three languages: Amharic, Ge'ez, and Tigrinya. Out of the total 1,847 words, Due to the limited expertise in proto-word reconstruction in Ethiopia, only a randomly selected subset of 39 words was provided to linguists for further verification and evaluation.

The verification process aimed to ensure that the reconstructed proto-words align with linguistic principles and historical reconstructions, thereby validating the computational approach applied to the dataset.

	GPT-4o
Precision	0.87
Recall	0.86
F1 Score	0.86
Accuracy	0.86

Table 5.4: Result of the verification data set by linguists

The result shows that the model achieves a strong and consistent performance at the character level, with an accuracy of 86% with an error rate of 14% indicating that a majority of the predicted characters align with the ground truth. The precision of 87% reflects the model's ability to make relevant predictions with minimal false positives. The recall of 86% highlights its capacity to capture most of the correct characters from the true sequences. The balanced F1 score of 86% further underscores the model's reliability in maintaining a good trade-off between precision and recall.

5.3 Ancestor word Reconstruction using Seq2Seq Models

The Main dataset was organized into annotated sequences specific to each language family divided into training, validation, and test sets. Models were trained on the respective training sets, tuned using validation sets, and evaluated on the test sets to measure their effectiveness in language reconstruction.

ParameterS	**LSTM-based NMT**	**GRU-based NMT**	**Transformer based NMT**
batch_size	1	1	1
beta1	0.9	0.9	-
beta2	0.999	0.999	-
dropout	0.27699	0.27699	0.1708861
embedding_size	80	489	128
epochs	100	80	120
eps	1e-08	1e-08	-
feedforward_dim	100	431	517
lr	0.000822	0.000122	0.00063
n_head	-	-	8
num_layers	1	1	3

Table 5.5: Parameters used for proposed model and baseline models.

The performance of GRU-based, transformer-based Neural Machine Translation (NMT) and LSTM-based models for language reconstruction tasks across three distinct language families are evaluated. The evaluation includes metrics such as accuracy, edit distance, and loss are used to assess model performance.

5.3.1 Accuracy

The **accuracy** of NMT-based models which are GRU-based and Transformer-based architectures was evaluated as a baseline for language reconstruction across three distinct language families: Chinese, Latin, and Ethio-Semitic. The performance of these baseline models was then compared to the proposed LSTM-based encoode-decode model providing a comprehensive comparison across these language families as shown in Figure 5.1.

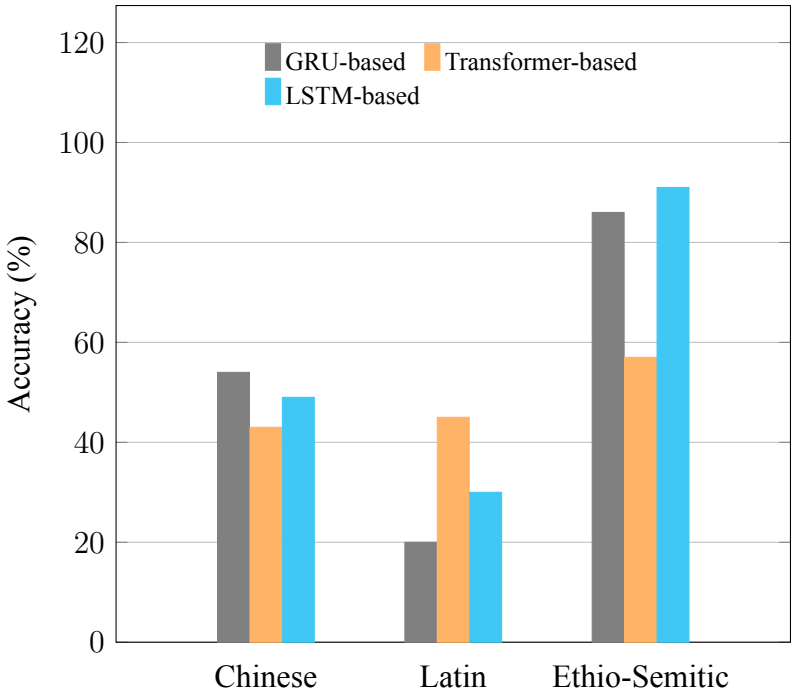

Figure 5.1: Accuracy comparison of baseline (GRU and Transformer NMT) and proposed (LSTM) encode-decode model for language reconstruction across three language families.

The accuaracy of the base line models and LSTM based model in proto-language reconstruction varied significantly across different language families. For the Chinese language family, the GRU-based model achieved the highest accuracy at 54%, followed by the LSTM-based model at 49%, and the Transformer-based model at 43%. Showing that the GRU-based model's architecture was particularly effective in capturing the linguistic patterns of Chinese.

The results for the Latin language family show a different pattern. The Transformer-based model proving to be the most effective, reaching an accuracy of 45%. Followed by the LSTM-based model with an accuracy of 30%, and the GRU-based model with 20% accuaracy. Showing that the Transformer's strong performance is largely due to its need for larger datasets. Its ability to manage long-range dependencies and process information simultaneously makes it especially well-suited to the Latin language family when there's sufficient data available to maximize these features.

For the Ethio-Semitic language family, the LSTM-based model proved to be the most effective with an accuracy of 91%. The superior performance of the LSTM-based model can be attributed to its strength in sequence learning and handling time dependencies even for small data set. The transformer based performed low for the Chinese and Ethio-Semitic language family due to the size of the data set used to train the model as it needs large data set to perform better.

5.3.2 Edit Distance

The second evalution metric is **edit distance**. It measures the number of changes (insertions, deletions, substitutions) needed to transform one string into another. Lower edit distances indicate higher similarity between the predicted proto-forms and the correct proto-forms.

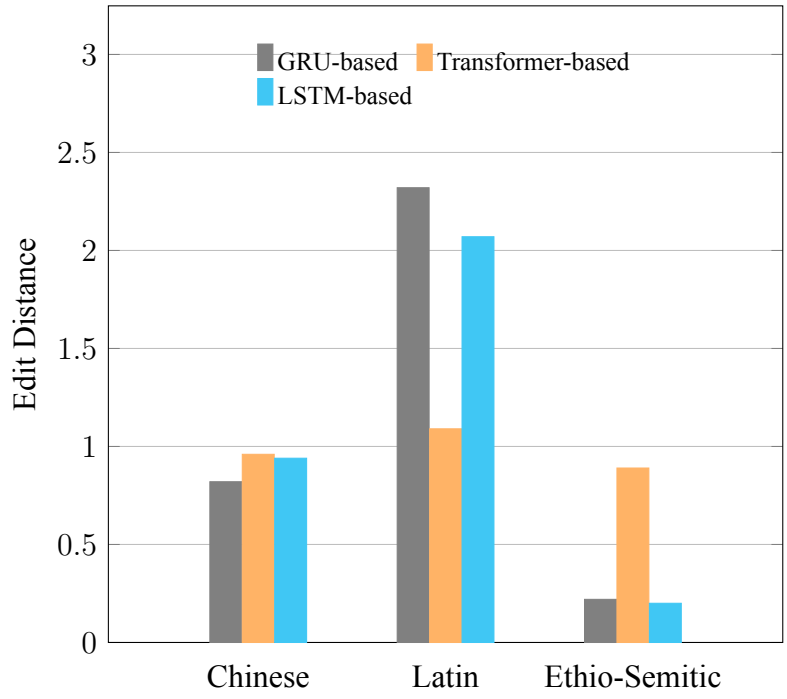

Figure 5.2: Edit Distance of different NMT-based models (GRU-based, Transformer-based) and LSTM-based encode-decode moodel for language reconstruction across three language families (Chinese, Latin, Ethio-Semitic).

The results of the edit distance analysis reveal significant insights into the performance of different neural network architectures across three language families: Chinese, Latin, and Ethio-Semitic. For Chinese, the GRU-based model stands out with the lowest edit distance of 0.82, indicating its ability to reconstruct proto-forms accurately. This suggests that the GRU architecture effectively captures the unique morphological and phonological characteristics of the Chinese language. In contrast, the Transformer-based model performs the least well in this context with an edit distance of 0.96, while the LSTM-based model falls in between at 0.94, reflecting its relatively moderate accuracy.

In the Latin language family, the performance dynamics shift, with the Transformer-based model achieving the lowest edit distance of 1.09. This can be attributed to the model's capacity to leverage large datasets, which is crucial for its performance. The LSTM-based model follows with a higher edit distance of 2.07, indicating a decline in accuracy, while the GRU-based model trails behind with an edit distance of 2.32, marking it as the least effective in this scenario. The results imply that the Transformer architecture is particularly well-suited to the Latin language, possibly due to its data-driven nature.

For Ethio-Semitic, the LSTM-based model emerges as the best performer with an impressive edit distance of 0.20. Its architectural advantages, such as capturing long-term dependencies and effectively managing sequential data, make it particularly adept at handling the complexities inherent in Ethio-Semitic languages. The GRU model closely follows with an edit distance of 0.22, demonstrating its strong performance as well. The Transformer-based model, however, lags behind with an edit distance of 0.89, indicating it is less suited to the intricacies of Ethio-Semitic languages.

Overall, these findings underscore the varying strengths of each model across different language families. The GRU model excels in Chinese due to its effective handling of the language's specific characteristics, while the Transformer model shines in Latin, benefitting from large datasets. The LSTM model proves to be the most versatile, successfully addressing the complexities of Ethio-Semitic languages. This analysis highlights the importance of selecting appropriate model architectures based on the unique linguistic features and available data for each language family, providing a roadmap for future research in language processing tasks.

5.3.3 Loss

The third evaluation metric we used is **loss**. Loss values in neural networks represent a measure of how well the model is performing: lower loss values indicate better performance.

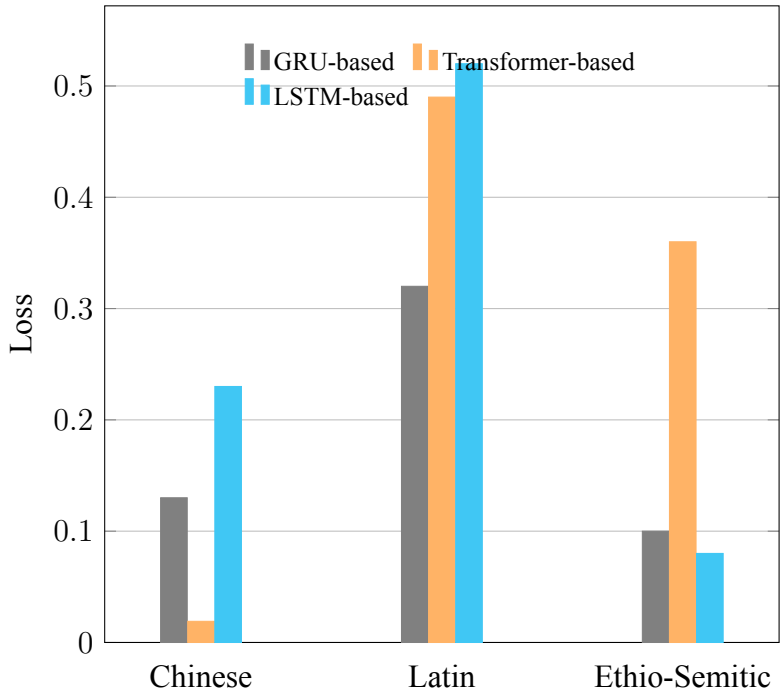

Figure 5.3: Loss of different NMT-based models (GRU-based, Transformer-based, LSTM-based) for language reconstruction across three language families (Chinese, Latin, Ethio-Semitic).

The evaluation of loss across different neural network architectures reveals critical insights into their performance during training for Chinese, Latin, and Ethio-Semitic languages. For Chinese, the Transformer-based model achieves the lowest loss of 0.019, demonstrating its superior capability to minimize training errors effectively. This positions the Transformer as the most proficient model for this language, suggesting that its architecture is well-suited to capture the intricate patterns within Chinese data. Following closely, the GRU-based model shows a loss of 0.13, indicating it performs reasonably well, albeit not as effectively as the Transformer. In contrast, the LSTM-based model exhibits the highest loss at 0.23, highlighting its relative inefficacy in this context.

In the Latin language family, the dynamics shift, with the GRU-based model achieving the lowest loss of 0.32. This suggests that it is particularly effective at minimizing errors during training for Latin, perhaps due to its ability to adapt to the linguistic characteristics of the data. The Transformer model follows with a loss of 0.49, indicating a less favorable performance, while the LSTM model has the highest loss at 0.52, reinforcing its position as the least effective model in this language family.

When examining Ethio-Semitic languages, both the GRU and LSTM models demonstrate commendable performance, achieving low losses of 0.1 and 0.08, respectively. This indicates their strong capabilities in handling the complexities of Ethio-Semitic languages. However, the Transformer-based model faces challenges, with a higher loss of 0.36, suggesting that it struggles more than the other architectures in minimizing training errors for this language family.

These varying accuracies emphasize that different language families have distinct structural complexities and historical developments, which make them more compatible with specific types of models. Therefore, when choosing an NMT model for proto-language reconstruction, it is crucial to consider the unique linguistic characteristics of the language family. Tailoring the model selection to the particular features of each language family can significantly enhance the accuracy and effectiveness of the reconstruction process.

Overall, these findings underscore the varying strengths of each model across different languages. The Transformer's effectiveness in Chinese contrasts with its struggles in Ethio-Semitic, while the GRU model consistently demonstrates robustness in both Latin and Ethio-Semitic contexts. The analysis emphasizes the importance of aligning model selection with the linguistic characteristics and error dynamics inherent to each language family, providing a clearer pathway for future research and application in language processing tasks as also the size of data set has an effect on performance of models.

Verfication for the three models

	GRU	Transformer	LSTM	By GPT-4o	linguists	Ge'ez	Tigriniya	Amharic
መሸፈት	səfiˀ	səfiˀ	səfiˀ	səfiˀ	səfiˀ	səfiˀ	miʃifaˀ	məʃəfəˀ
መፍርጃ	məħiɬəri	mħiɬəri	məħiɬər	məħiɬər	məħiɬəri	məħiɬəri	məħiɬəri	məfərja
ሀከከ	tətʃ͡ʼək'ə	tətatʃ͡ʼək'a	tətʃ͡ʼək'a	tətʃ͡ʼək'a	tətak'a	tətagomə	tətʃ͡ʼək'a	həkəkə
አሰሰ	gəsəsə	gəsəsi	gəsəsə	gəsəsə	gəsəsə	gəsəsə	həsəsɨ	ʔəsəsə
ሸመተ	ʃɨmətə	məsətə	ʃəmətə	ʃəmətə	ʃəmətə	məsədə	ʃəmətə	ʃɨmətə
በዛ	rəbəħə	bəz	bəziħə	rəbɨħə	bəziħə	rəbəħə	beziħə	bəza
አቀጣጠለ	ʔəqet͡sʼə	bəqet͡sʼə	ʔəqat͡sʼə	ʔəqat͡sʼə	ʔəqat͡sʼə	bəqet͡sʼə	ʔəqat͡sʼələ	ʔək'əˀat'ələ

Table 5.6: Examples of patterns from the Three models and the linguist reconstructed proto-forms.

For verification 39 cognate words were given for lingusts from the main dataset and we compared the result with the output of the three methods. When comparing the GRU based NMT with the linguists proto-word reconstruction, the changes indicate that the prediction process often altered vowels and some consonants, particularly in complex clusters. The most frequent changes involved vowel shifts (e.g., /a/ to /ə/, /ɨ/ to /ə/, /e/ to /u/) and consonant substitutions (e.g., /b/ to /r/,/k/ to /m/).

On the other hand when comparing the transformer based NMT with the lingusts data, the results indicate that the model often confuses central vowels like /ə/ and /ɨ/ as well as interchanges vowels /a/ and /e/ Consonant substitutions also occur, particularly with glottal, pharyngeal, and liquid sounds, where distinctions between /m/, /ʃ and /r/ are problematic. Despite these challenges, sounds like /s/, /t/, /d/, and /q/ are consistently predicted accurately, highlighting their stability in the model's output.

For the LSTM based NMT, the character-by-character comparison shows that the model accurately predicts most characters in the majority of cases, with consistent success in predicting sounds like /s/, /ə/, /t/, and /m/ However, there are notable errors, especially in distinguishing between similar vowel sounds such as /ə/ and /ɨ/ as well as in predicting consonants like /k/ , /r/, and /q/.

In addition we compared the 39 proto-words reconstructed by the linguist with generated syntetic data. It reveals a mix of correct and incorrect predictions across various rows. Most predictions matched the expected sounds, such as /s/, /ə/ , /f/, /i/, /t/, /m/, /ħ/, and others, indicating a solid performance for these characters. However, several specific mismatches were noted, particularly where the expected /a/ was incorrectly predicted as /ˉˉ/, /k/ as /ʃ/, and another /a/ as /ə/. Also showed errors, with /b/ predicted as /r/ and /z/ as /b/ Similar issues appeared where /a/ was replaced with /q/ and /q/ as /k/ and /ɨ/ as /ə/.

As demonstrated in Table 5.7, the proposed method significantly outperforms the baseline models according to the linguists' evaluations for the Ethio-Semitic language family. This performance is achieved despite an error rate associated with the GPT-4o synthetic dataset, on which the model was trained. These results indicate that our approach offers a more accurate and reliable solution compared to existing models, even when trained with synthetic data.

	GRU based NMT	Transformer based NMT	LSTM based NMT
Precision	0.70	0.68	0.79
Recall	0.71	0.68	0.78
F1 Score	0.69	0.67	0.77
Accuracy	0.71	0.68	0.78

Table 5.7: The performance of the three models comparing with the linguists verfication

The two research questions mentioned in Section 1.1 are answered as follows:

RQ1 How effective are large language models (LLMs) at generating high-quality synthetic cognate datasets for the Ethio-Semitic language family, particularly when using in-context learning to address data scarcity and phonological uniqueness?

As seen in Section 5.2, with 11 examples for in-context learning, GPT-4o was able to correctly predict the proto-form 85% of the time for the Gold Dataset. The high accuracy rate suggests that GPT-4o can be used to generate the proto-form for the Main Dataset with an error margin of 0.094.

RQ2 To what extent can an LSTM-based encode-decode model, trained on both generated synthetic data and existing linguistic data, accurately reconstruct proto-words of Ethio-Semitic languages, compared to existing GRU and transformer models?

Using the proto-forms that were generated by GPT-4o, Seq2Seq models, including popular Machine Translation (MT) architectures were trained and tested for constructing the proto-form of the Main Dataset. The results show that the LSTM-based encode-decode model achieved the highest accuracy of all the other models in the experiment. Additionally, the performance of the Ethio-Semitic language family was higher than that of the other two languages (Chinese and Latin). In the comparison with other languages, it was also observed that there was no single architecture that performed better across all languages: the GRU-based model had the best performance for Chinese, the Transformer-based model had the best performance for Latin, and the LSTM-based model had the best performance for Ethio-Semetic. Hence, whether or not a single model can effectively reconstruct proto-forms for all languages remains inconclusive.

5.4 Summary

In this chapter, the results of our experiment aimed at reconstructing proto-words have been presented and analyzed. Our methodology involved a series of structured steps to extract and preprocess data from the identified cognates across the given language datasets. Using the proposed proto-word reconstruction method, a new dataset was successfully generated from cognate words to their proto-forms. The effectiveness of the reconstructed proto-words was further evaluated by comparing them against expert linguistic reconstructions. To validate the accuracy of the generated proto-forms, a random subset of reconstructed proto-words was reviewed by linguists. The assessment showed an accuracy of 87%, indicating a high level of agreement between the reconstructed proto-forms and expert reconstructions. Using this data set the LSTM-based encode-decode model demonstrating an accuracy of 91%.

Chapter 6

Conclusion and Future work

As languages change and evolve, fields such as historical linguistics examine the patterns of these changes. One task in historical linguistics is historical language reconstruction: the process of recreating ancestral language forms from their present-day descendants. A series of experiments were conducted to reconstruct the proto-form for three Ethio-Semitic languages: Amharic, Tigrinya, and Ge'ez. Data were collected from a three-way dictionary, and (1) linguists were tasked with constructing the proto-form for 74 cognates, while (2) automated methods were employed to identify cognates from the remainder of the datasets. In-context learning was then utilized to generate synthetic protoforms for words in the cognate dataset, with a subset of the generated protoforms being verified by linguists. Three different Seq2Seq models were trained and tested for proto-form reconstruction using the dataset. It was found that GPT-4 could generate proto-forms for the dataset with an error margin of 0.094 and an accuracy of 85% on the Gold dataset created by linguists. Among the Seq2Seq models, the LSTM-based model outperformed the other two, achieving an accuracy of 91%.

Although the effectiveness of in-context learning for historical language reconstruction in the three languages is demonstrated by our research, there are limitations due to the lack of human experts to validate the full dataset. To verify the output of GPT-4o 39 words out of 1847 were given to linguists for evaluation, resulting in an accuracy of 86% with an error rate of 14%. In the future, human evaluation will be extended to the full dataset for more robust results. Additionally, the research focuses only on three of the seven languages in the family. The inclusion of more languages would allow for the capture of additional patterns and the generation of proto-forms that better reflect the ancestral language. However, with these limitations in mind, the research contributes (1) a dataset of 1847 automatically detected cognates and 74 human expert-identified cognates in the three languages, (2) 1847 proto-forms automatically generated by GPT-4o and 74 proto-forms constructed by human experts, and (3) benchmark results of models on both automatic and human-expert created datasets. Overall, this research is considered a starting point for further investigation into historical language reconstruction for Ethio-Semitic languages.

For future work, it is essential to expand human evaluation to the full dataset of reconstructed proto-words. Additionally, the current reconstruction focuses only on three languages within the Ethio-Semitic language family. Including the remaining four languages would help capture additional linguistic patterns and enhance the accuracy of the reconstruction.

References

[1] Encyclopaedia Britannica. Language. https://www.britannica.com/topic/language, n.d. Accessed: 2024-07-30.

[2] Merriam-Webster. Language. https://www.merriam-webster.com/dictionary/language, n.d. Accessed: 2024-07-30.

[3] Theodora Bynon. *Historical linguistics*. Cambridge University Press, 1977.

[4] Hans Henrich Hock. An introduction to historical and comparative linguistics, 1976.

[5] Hedvig Skirgård. Disentangling ancestral state reconstruction in historical linguistics: Comparing classic approaches and new methods using oceanic grammar. *Diachronica*, 2024.

[6] Khuyagbaatar Batsuren, Gábor Bella, Fausto Giunchiglia, et al. Cognet: A large-scale cognate database. In *ACL 2019 The 57th Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference*, pages 3136–3145. Association for Computational Linguistics, 2019.

[7] Johann-Mattis List. Automatic inference of sound correspondence patterns across multiple languages. *Computational Linguistics*, 45(1):137–161, 2019.

[8] Mikel Santesteban and Albert Costa. Are cognate words "special"?". *Cognitive control and consequences of multilingualism*, 2:97, 2016.

[9] Carlo Meloni, Shauli Ravfogel, and Yoav Goldberg. Ab antiquo: Neural proto-language reconstruction. *arXiv preprint arXiv:1908.02477*, 2019.

[10] Young Min Kim, Kalvin Chang, Chenxuan Cui, and David Mortensen. Transformed protoform reconstruction. *arXiv preprint arXiv:2307.01896*, 2023.

[11] Robert Hetzron, Alan S Kaye, and Ghil'ad Zuckermann. Semitic languages. In *The World's Major Languages*, pages 568–576. Routledge, 2018.

[12] Leonid Kogan. Common origin of ethiopian semitic: the lexical dimension. *Scrinium*, 1(1):367–396, 2005.

[13] Hellina Nigatu, Atnafu Tonja, and Jugal Kalita. The less the merrier? investigating language representation in multilingual models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12572–12589, Singapore, December 2023. Association for Computational Linguistics.

[14] 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.

[15] Arkil Patel, Siva Reddy, Dzmitry Bahdanau, and Pradeep Dasigi. Evaluating in-context learning of libraries for code generation. *arXiv preprint arXiv:2311.09635*, 2023.

[16] Stillman Translations. Language functions of language: Definition of language translations, 2023. Accessed: 2024-10-11.

[17] University of Pennsylvania. Language change, 2003. Accessed: 2024-10-11.

[18] Robin S Chapman. Language development in children and adolescents with down syndrome. *The handbook of child language*, pages 641–663, 1996.

[19] Sarah Grey Thomason. Linguistic areas and language history. In *Languages in contact*, pages 311–327. Brill, 2000.

[20] Michael Pleyer. The role of interactional and cognitive mechanisms in the evolution of (proto)language(s). *Lingua*, 282:103458, 2023.

[21] Przemyslaw Zywiczynski, Nathalie Gontier, and Slawomir Wacewicz. The evolution of (proto-)language: Focus on mechanisms. *Language Sciences*, 63:1–11, 2017. Language Evolution: Focus on Mechanisms.

[22] Gerhard Jäger. Computational historical linguistics. *Theoretical Linguistics*, 45(3-4):151–182, 2019.

[23] Alexandre Bouchard-Côté, David Hall, Thomas L Griffiths, and Dan Klein. Automated reconstruction of ancient languages using probabilistic models of sound change. *Proceedings of the National Academy of Sciences*, 110(11):4224–4229, 2013.

[24] Alina Maria Ciobanu and Liviu P Dinu. Ab initio: Automatic latin proto-word reconstruction. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1604–1614, 2018.

[25] Henry M Hoenigswald. *Comparative method, internal reconstruction, typology*. na, 1992.

[26] Anna Giacalone Ramat and Paolo Ramat. *The Indo-European Languages*. Routledge, 2015.

[27] Gábor Takács. New lexical materials for the proto-afro-asiatic anatomical and physiological terminology i: Body part names with initial labials: General terms, head and neck. *Lingua Posnaniensis*, 64(2):107–144, 2022.

[28] Zygmunt Frajzyngier. Afroasiatic languages. In *Oxford research encyclopedia of linguistics*. 2018.

[29] Rainer Voigt. The afroasiatic languages.(cambridge language surveys), 2017.

[30] John Huehnergard et al. *The Semitic Languages*. Routledge, 2013.

[31] EKI Archive. Rom2-ti document. https://arhiiv.eki.ee/wgrs/v2_2/rom2_ti.pdf, n.d. Accessed: 2024-07-30.

[32] BA PRABHAKAR BABU. The international phonetic alphabet (ipa). *Encyclopaedia of the Linguistic Sciences: Issues and Theories*, page 297, 2008.

[33] Wikipedia contributors. International phonetic alphabet. https://en.wikipedia.org/wiki/International_Phonetic_Alphabet, n.d. Accessed: 2024-07-30.

[34] PyPI contributors. Epitran. https://pypi.org/project/epitran/, n.d. Accessed: 2024-07-30.

[35] David R Mortensen, Siddharth Dalmia, and Patrick Littell. Epitran: Precision g2p for many languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[36] GitHub contributors. Epitran. https://github.com/dmort27/epitran/tree/master, n.d. Accessed: 2024-07-30.

[37] Lyle Campbell. Historical linguistics: An introduction. http://tscheer.free.fr/scan/Campbell%2098%20-%20Historical%20Linguistics.%20An%20Introduction.pdf, 1998. Accessed: 2024-07-30.

[38] Henry M Hoenigswald. The comparative method. *Current trends in Linguistics*, 2:51–62, 1973.

[39] LingPy Team. File formats and extensions. https://lingpy.org/tutorial/formats.html, n.d. Accessed: 2024-07-30.

[40] Turchin Peter, Peiros Ilia, and Gell-Mann Murray. Analyzing genetic connections between languages by matching consonant classes. *Вопросы языкового родства*, (5 (48)):117–126, 2010.

[41] A. Dolgopolsky. Gipoteza drevnejšego rodstva jazykovych semej severnoj evrazii s verojatnostej točky zrenija [a probabilistic hypothesis concering the oldest relationships among the language families of northern eurasia], 1964.

[42] Johann-Mattis List. Sca: Phonetic alignment based on sound classes. In *European Summer School in Logic, Language and Information*, pages 32–51. Springer, 2010.

[43] Johann-Mattis List. LexStat: Automatic detection of cognates in multilingual wordlists. In Miriam Butt, Sheelagh Carpendale, Gerald Penn, Jelena Prokić, and Michael Cysouw, editors, *Proceedings of the EACL 2012 Joint Workshop of LINGVIS & UNCLH*, pages 117–125, Avignon, France, April 2012. Association for Computational Linguistics.

[44] James Clackson. *Indo-European linguistics: an introduction*. Cambridge university press, 2007.

[45] Hang Li. Deep learning for natural language processing: advantages and challenges. *National Science Review*, 5(1):24–26, 2018.

[46] Julia Hirschberg and Christopher D Manning. Advances in natural language processing. *Science*, 349(6245):261–266, 2015.

[47] Peter Dekker. Reconstructing language ancestry by performing word prediction with neural networks. *Master. Amsterdam: University of Amsterdam*, 2018.

[48] Risto Miikkulainen, Jason Liang, Elliot Meyerson, Aditya Rawal, Dan Fink, Olivier Francon, Bala Raju, Hormoz Shahrzad, Arshak Navruzyan, Nigel Duffy, et al. Evolving deep neural networks. In *Artificial intelligence in the age of neural networks and brain computing*, pages 269–287. Elsevier, 2024.

[49] Rene Y Choi, Aaron S Coyner, Jayashree Kalpathy-Cramer, Michael F Chiang, and J Peter Campbell. Introduction to machine learning, neural networks, and deep learning. *Translational vision science & technology*, 9(2):14–14, 2020.

[50] Jie Yang and Jun Ma. Feed-forward neural network training using sparse representation. *Expert Systems with Applications*, 116:255–264, 2019.

[51] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*, 33(12):6999–7019, 2021.

[52] Ian Goodfellow. Deep learning, 2016.

[53] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[54] Yiming Zhang, Shi Feng, and Chenhao Tan. Active example selection for in-context learning. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9134–9148, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[55] Haoyu Liu, Jianfeng Liu, Shaohan Huang, Yuefeng Zhan, Hao Sun, Weiwei Deng, Furu Wei, and Qi Zhang. se^2: Sequential example selection for in-context learning. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics ACL 2024*, pages 5262–5284, Bangkok, Thailand and virtual meeting, August 2024. Association for Computational Linguistics.

[56] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*, 2022.

[57] Vladimir I Levenshtein et al. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union, 1966.

[58] Johann-Mattis List, Simon J Greenhill, and Russell D Gray. The potential of automatic word comparison for historical linguistics. *PloS one*, 12(1):e0170046, 2017.

[59] Roddy MacSween and Andrew Caines. An expectation maximisation algorithm for automated cognate detection. In Raquel Fernández and Tal Linzen, editors, *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 476–485, Online, November 2020. Association for Computational Linguistics.

[60] Johann-Mattis List, Robert Forkel, and Nathan W Hill. A new framework for fast automated phonological reconstruction using trimmed alignments and sound correspondence patterns. *arXiv preprint arXiv:2204.04619*, 2022.

[61] VSDS Akavarapu and Arnab Bhattacharya. Automated cognate detection as a supervised link prediction task with cognate transformer. *arXiv preprint arXiv:2402.02926*, 2024.

[62] Alexandre Bouchard-Côté, Thomas L Griffiths, and Dan Klein. Improved reconstruction of protolanguage word forms. In *Proceedings of human language technologies: The 2009 annual conference of the north american chapter of the association for computational linguistics*, pages 65–73, 2009.

[63] Alina Maria Ciobanu, Liviu P Dinu, and Laurentiu Zoicas. Automatic reconstruction of missing romanian cognates and unattested latin words. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3226–3231, 2020.

[64] Carlo Meloni, Shauli Ravfogel, and Yoav Goldberg. Ab antiquo: Neural protolanguage reconstruction. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4460–4473, Online, June 2021. Association for Computational Linguistics.

[65] Remo Nitschke. Restoring the sister: Reconstructing a lexicon from sister languages using neural machine translation. In Manuel Mager, Arturo Oncevay, Annette Rios, Ivan Vladimir Meza Ruiz, Alexis Palmer, Graham Neubig, and Katharina Kann, editors, *Proceedings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas*, pages 122–130, Online, June 2021. Association for Computational Linguistics.

[66] Andre He, Nicholas Tomlin, and Dan Klein. Neural unsupervised reconstruction of protolanguage word forms. *arXiv preprint arXiv:2211.08684*, 2022.

[67] Chenxuan Cui, Ying Chen, Qinxin Wang, and David R Mortensen. Neural protolanguage reconstruction. *arXiv preprint arXiv:2404.15690*, 2024.

[68] Aristides Milios, Siva Reddy, and Dzmitry Bahdanau. In-context learning for text classification with many labels. In *Proceedings of the 1st GenBench Workshop on (Benchmarking) Generalisation in NLP*, pages 173–184, 2023.

[69] Aleksandra Edwards and Jose Camacho-Collados. Language models for text classification: Is in-context learning enough? *arXiv preprint arXiv:2403.17661*, 2024.

[70] Tianle Li, Xueguang Ma, Alex Zhuang, Yu Gu, Yu Su, and Wenhu Chen. Few-shot in-context learning for knowledge base question answering. *arXiv preprint arXiv:2305.01750*, 2023.

[71] Yunlong Chen, Yaming Zhang, Jianfei Yu, Li Yang, and Rui Xia. In-context learning for knowledge base question answering for unmanned systems based on large language models. In *China Conference on Knowledge Graph and Semantic Computing*, pages 327–339. Springer, 2023.

[72] Kassian Alexei, Starostin George, Dybo Anna, and Chernov Vasiliy. The swadesh wordlist. an attempt at semantic specification. *Вопросы языкового родства*, (16 (59)):46–89, 2010.

[73] Wikipedia contributors. Swadesh list. https://en.wikipedia.org/wiki/Swadesh_list, n.d. Accessed: 2024-07-30.

[74] zra dawit adhana. *Lsanat Sem*. Kidst Slase Menfesawi Kolej, addis ababa, Year of Publication.

[75] Peter Christen and Peter Christen. Data pre-processing. *Data matching: Concepts and techniques for record linkage, entity resolution, and duplicate detection*, pages 39–67, 2012.

[76] Arthur D Chapman. *Principles and methods of data cleaning*. GBIF, 2005.

[77] Isabelle Guyon and André Elisseeff. An introduction to feature extraction. In *Feature extraction: foundations and applications*, pages 1–25. Springer, 2006.

[78] Lincoln A. Mullen, Kenneth Benoit, Os Keyes, Dmitry Selivanov, and Jeffrey Arnold. Fast, consistent tokenization of natural language text. *Journal of Open Source Software*, 3(23):655, 2018.

[79] Sabrina J Mielke, Zaid Alyafeai, Elizabeth Salesky, Colin Raffel, Manan Dey, Matthias Gallé, Arun Raja, Chenglei Si, Wilson Y Lee, Benoît Sagot, et al. Between words and characters: A brief history of open-vocabulary modeling and tokenization in nlp. *arXiv preprint arXiv:2112.10508*, 2021.

[80] Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, 35:30583–30598, 2022.

[81] Muhammad Usman Hadi, Rizwan Qureshi, Abbas Shah, Muhammad Irfan, Anas Zafar, Muhammad Bilal Shaikh, Naveed Akhtar, Jia Wu, Seyedali Mirjalili, et al. Large language models: a comprehensive survey of its applications, challenges, limitations, and future prospects. *Authorea Preprints*, 2023.

[82] Ting-Yun Chang and Robin Jia. Data curation alone can stabilize in-context learning. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8123–8144, Toronto, Canada, July 2023. Association for Computational Linguistics.

[83] Jiuhai Chen, Lichang Chen, Chen Zhu, and Tianyi Zhou. How many demonstrations do you need for in-context learning?, 2023.

[84] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.

[85] Wentong Chen, Yankai Lin, ZhenHao Zhou, HongYun Huang, Yantao Jia, Zhao Cao, and Ji-Rong Wen. Icleval: evaluating in-context learning ability of large language models. *arXiv preprint arXiv:2406.14955*, 2024.

[86] 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.

[87] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[88] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arxiv. *arXiv preprint arXiv:1810.04805*, 2019.

[89] BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*, 2022.

[90] Ahmet Üstün, Viraat Aryabumi, Zheng-Xin Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, et al. Aya model: An instruction finetuned open-access multilingual language model. *arXiv preprint arXiv:2402.07827*, 2024.

[91] Jiuhai Chen, Lichang Chen, Chen Zhu, and Tianyi Zhou. How many demonstrations do you need for in-context learning? *arXiv preprint arXiv:2303.08119*, 2023.

[92] Dzmitry Bahdanau, Dmitriy Serdyuk, Philemon Brakel, Nan Rosemary Ke, Jan Chorowski, Aaron Courville, and Yoshua Bengio. Task loss estimation for sequence prediction. *arXiv preprint arXiv:1511.06456*, 2015.

[93] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[94] Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404:132306, 2020.

[95] Analytics Vidhya. 11 important model evaluation metrics for data science projects. https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-metrics/, 2019. Accessed: 2024-07-30.