UNIVERSITY OF GOTHENBURG

DEPARTMENT OF PHILOSOPHY,
LINGUISTICS AND THEORY OF SCIENCE

Breaking Barriers: Enhancing Universal Dependency Parsing for Amharic

Advancing NLP for A Low-Resource Language

Dawit Jembere

Essay/Thesis:	Master's thesis, 30 credits
Programme/Course:	Master in Language Technology, 120 credits
Level:	Second cycle
Semester:	Spring, 2025
Supervisor:	Arianna Masciolini and Abraham Yeshalem
Examiner:	Aarne Ranta
Keywords:	Language Technology

Abstract

This study advances Amharic dependency parsing by expanding and refining the existing Universal Dependencies (UD) Treebank (Seyoum, Miyao, and Mekonnen, 2018). As a morphologically rich and under-resourced language, Amharic poses unique challenges in natural language processing (NLP), particularly in syntactic and morphological parsing. Leveraging the UD framework and the transformer-based toolkit, Trankit, this work achieves improved parsing accuracy, outperforming the results obtained with UDPipe and Turku models by Seyoum, Miyao, and Mekonnen (2020) across multiple evaluation metrics. This result demonstrates that dataset augmentation, coupled with rigorous syntactic validation, can substantially enhance parsing performance and offer a scalable pathway for NLP development in low-resource languages.

Keywords: Amharic, Universal Dependencies, Low-Resource Language, Tokenization, Dependency Parsing, Treebank Expansion, Natural Language Processing

Acknowledgements

> “በፍጹም ልብህ በእግዚአብሔር ታመን፤
> በራስህ ማስተውል አትደገፍ፤
> በመንገድህ ሁሉ እርሱን እወቅ፤
> እርሱም ጎዳናህን ቀና ያደርገዋል። ”
> — ምሳሌ 3፥ 5–6

First and foremost, I give thanks to Almighty God for His grace, guidance, and strength throughout this journey. Without His wisdom and provision, this work would not have been possible.

I would like to express my deepest gratitude to my advisors, Arianna Masciolini and Abraham Yeshalem, for their guidance and encouragement. I am especially grateful to Arianna for her dedication, insightful feedback, and thoughtful discussions, all of which greatly enriched the development of this thesis.

Finally, I thank my family for their unwavering love, patience, and support throughout this academic journey.

Preface

This thesis stems from the enduring fascination with natural language processing (NLP) and a commitment to addressing the technological inequalities faced by under-resourced languages. Focusing on Amharic-a morphologically rich language central to Ethiopia's cultural and communicative landscape-this work emerged from a desire to bridge the gap between theoretical linguistics and practical language technology. Motivated by the scarcity of resources and tools available for morphologically rich, low-resource languages, I set out to explore more effective approaches to syntactic and morphological analysis using NLP techniques.

The idea for this research took shape during the 1st UniDive Training School, held at the Technical University of Moldova, Chisinau, in July 2024. Several enlightening discussions with Arianna Masciolini, along with the comprehensive training sessions, played a key role in inspiring and guiding the direction of this work.

This project enhances Amharic dependency parsing by expanding and refining the Universal Dependencies (UD) Treebank (Seyoum et al., 2018) and adopting Trankit, a transformer-based multilingual parser (Nguyen et al., 2021). It combines theoretical foundations with practical implementations, contributing to the broader effort to advance language technology for Ethiopian languages. Structured to mirror this investigative process, the thesis begins by introducing the motivation and aims (Chapter 1), contextualizes Amharic's linguistic and technological landscape and reviews previous works in dependency parsing (Chapter 2), details data curation and methodology (Chapter 3), presents findings and discussion (Chapter 4), and concludes by outlining key contributions and directions for future research (Chapter 5).

This work represents not only an academic milestone but also a personal journey of technical growth and intellectual discovery.

Contents

List of Figures

List of Tables

1 Introduction

The field of Natural Language Processing (NLP) has made significant progress in enabling machines to understand and generate human language (Jurafsky and Martin, 2021). Parsing, the computational process to determine the syntactic structure of a sentence according to formal grammatical rules (Jurafsky and Martin, 2021), is foundational to natural language processing (NLP). A key task in this domain is dependency parsing, a specific approach that identifies grammatical relationships between words to map hierarchical structures (Manning and Schutze, 1999). For morphologically rich languages like Amharic, dependency parsing faces unique challenges due to complex inflectional systems and limited annotated resources.

Dependency parsing is fundamental to interpreting natural language, enabling machines to analyze and interpret text in a structured manner, an essential step in the transformation of unstructured text into structured and interpretable data. In many NLP tasks – such as machine translation, question answering, and information extraction – accurate parsing is essential for ensuring high performance, as it enables machines to grasp syntactic relationships between words.

For languages like Amharic – a Semitic language characterized by complex morphology and root-and-pattern verb paradigms (Leslau, 1995) – dependency parsing presents challenges. These complex syntactic structures and rich morphological variations require careful handling, making parsing particularly demanding. In addition to its morphosyntactic complexity, there is a need for expansion of the treebank, greater attention to annotation quality, and adaptation of state-of-the-art models capable of effectively capturing these intricacies.

Despite various efforts to train parsers and improve performance, their accuracy remains insufficient. The Amharic Universal Dependencies (UD) Treebank (Seyoum, Miyao, and Mekonnen, 2018), developed as a crucial resource for dependency parsing, offers a basis for analyzing this language's morphosyntax. Previous efforts have also highlighted persistent gaps in parsing accuracy, particularly when dealing with Amharic's morphological complexity and long-range syntactic dependencies.

The goal of this project is to improve parsing accuracy and usability for Amharic, a low-resource language with complex morphology and syntax. To achieve this, the work proceeds in three main stages. First, it involves expanding and refining the Amharic UD Treebank by addressing annotation inconsistencies and increasing the dataset's size and representativeness. Second, a complete parsing pipeline is trained from scratch using Trankit, a modern transformer-based toolkit designed for multilingual NLP. This pipeline includes a sentence segmenter, tokenizer, multiword token (MWT) expander, part-of-speech tagger, lemmatizer, morphological analyzer, and dependency parser, enabling it to process raw text directly without the need for pre-processing. Third, the resulting model is intended to be made publicly available, providing researchers and developers with an accessible, ready-to-use tool for Amharic language processing. By combining data enhancement with state-of-the-art modeling and open access, this project contributes to advancing NLP support for Amharic and other typologically diverse, low-resource languages.

1.1 Research Questions

This thesis explores the following research questions:

- Does the use of transformer-based (Trankit) models lead to improved parsing accuracy for Amharic compared to prior toolkits (e.g., UDPipe)**?**
- What is the impact of expanding and refining the Amharic UD Treebank on the performance of parsing pipeline tasks (e.g., tokenization, dependency parsing)**?**

1.2 Motivation and Real-World Applications

Amharic, the official working language of Ethiopia, is the second most widely spoken Semitic language after Arabic. It is spoken by tens of millions and serves as a lingua franca across several regional states, including Addis Ababa, Amhara, Central, South, Southwest, Benishangul-Gumuz, and Gambella. Based on Ethiopia's 2007 population census (which reported 73.9 million people) and subsequent projections, the country's total population is estimated to exceed 118 million by 2024 (Central Statistical Agency of Ethiopia, 2007; World Bank, 2024).

Despite its cultural and economic importance, Amharic has been largely underrepresented in the field of computational linguistics. The language's complex features –like rich inflection, subject-object agreement, and clitics-have been shown to present a unique challenge for dependency parsers, which identifies relationships between words in a sentence. A well-optimized dependency parser can still play a critical role in enhancing NLP applications – such as machine translation, speech recognition, and semantic analysis – especially for low-resource languages, where end-to-end models often struggle due to limited annotated data. In such contexts, accurate syntactic analysis can provide essential structural insights that boost overall system performance.

The primary motivation for this project is to bridge the resource gap in Amharic NLP, particularly the lack of large-scale, high-quality annotated datasets. By improving the Amharic UD Treebank and leveraging modern parsing technologies, this work contributes to the development of inclusive NLP resources, enabling better support for low-resource languages.

1.3 Contributions

This project makes several key contributions to Amharic NLP. First, it builds a customized pipeline featuring a transformer-based dependency parser that will be available for downstream applications. Second, it extends the Amharic UD treebank to cover a wide range of linguistic phenomena, particularly verb morphology, cliticization, and syntactic agreement. Finally, this work contributes to open-source NLP resources by making the enhanced treebank and trained model publicly available (the model is available on GitHub [1] until its release through the official Trankit website[2], fostering further research and development in Amharic NLP.

1.4 Scope

This thesis focuses on improving dependency parsing for Amharic within the context of the Universal Dependencies (UD) framework. It builds on existing resources, particularly the Amharic UD Treebank. While transformer-based parsing models like Trankit, a Light-Weight Transformer-based toolkit for Multilingual Natural Language Processing (Van Nguyen et al., 2021) are utilized, the project aims to develop full pipelines and concentrates on dependency parsing as a fundamental task. Although this research is specific to Amharic, the approaches and methodologies explored could be adapted to other under-resourced languages.

1.5 Structure of the Thesis

This thesis is organized to provide a comprehensive exploration of dependency parsing for Amharic, from foundational concepts to experimental results and future research directions. Chapter 2 lays the groundwork for the thesis by presenting essential background on the Amharic language, and the Universal Dependecies Framework. It reviews relevant previous works in Amharic, emphasizing the application of the UD framework and the particular challenges associated with parsing morphologically rich languages such as Amharic (Seyoum, Miyao, and Mekonnen, 2020; Seyoum, Miyao, and Mekonnen, 2018; Seyoum, Miyao, and Mekonnen, 2016; Degu and Gebeyehu, 2022). Chapter 3 details the data and methodology, including

[1] https://github.com/Jembda/MLT-Thesis-Trankit

[2] https://trankit.readthedocs.io/en/latest/performance.html

data collection, model development, and evaluation techniques, providing insights into the experimental setup. Chapter 4 presents results and discussion, showcasing model performance improvements in Labled Attachment Score (LAS) and Unlabeled Attachment Score (UAS); finally presents a discussion of the findings, and their significance for Amharic NLP. Chapter 5 concludes the thesis by summarizing key contributions and highlighting areas for further exploration, including potential enhancements to Amharic parsing models and the adaptation of these techniques to other low-resourced languages.

2 Background

Building on the motivation and objectives presented in Chapter 1, this chapter establishes the linguistic and computational foundations essential for understanding the process of dependency parsing. It begins by briefly introducing the structural characteristics of the language – its rich morphology, cliticization, and orthographic inconsistencies – which significantly impact syntactic analysis. It then introduces the Universal Dependencies (UD) framework as a standardized approach for cross-linguistic syntactic annotation, highlighting its relevance for low-resource languages. A review of dependency parsing techniques follows, emphasizing recent dependency parsing achievements, and it summarizes prior related works in Amharic, particularly the creation and use of UD treebanks, which inform the methodological choices made in the study. This foundation sets the stage for Chapter 3, where I detail the data sources and tools employed in parser training and experiments.

2.1 The Language

Amharic is the official working language of Ethiopia and the second most widely spoken Semitic language after Arabic. Its writing system is derived from the ancient Ge'ez script, consisting of 33 base characters, each with seven vowel forms or "orders" reflecting the language's seven-vowel system (Demeke, 2017). Written from left to right, modern Amharic orthography encodes syllables as consonant-vowel (CV) pairs. In Amharic, nouns are inflected to express gender, case (**ገረድዋን** gäräd-wa-n maid-her-Acc "her maid"), definiteness (e.g., **ብዛቱ** bizat-u amount-def "the amount"), and number (e.g., **በወንዶች** bä-wänd-occ by-man-pl "by men"). The language employs two primary plural markings: -occ and innä- (Demeke, 2017, pp. 93–94). Compound word writing (e.g., separated by a space, as in **ዳቦ ቤት** "bakery" (**ዳቦ** "bread" and **ቤት** "house"), separated by a hyphen, as in **ስነ-ልቦና** "psychology" (**ስነ** "art", "science" or "discipline of" and **ልቦና** "inner"); or as one word, **ቤተመፅሀፍት** "library" (**ቤተ** "house" and **መፅሀፍት** "books")) leads to considerable variation in written forms (Seyoum, Miyao, and Mekonnen, 2020; Seyoum, Miyao, and Mekonnen, 2018; Seyoum, Miyao, and Mekonnen, 2016).

These morphological and orthographic complexities intersect with high morphological variability to create challenging terrain for computational analysis. Besides, Amharic is a morphologically rich language in the sense that grammatical relations such as subject, object, and syntactic roles are typically encoded at the word level through inflectional morphology rather than word order (Habash, 2010; Seyoum, Miyao, and Mekonnen, 2016). As in Semitic languages more broadly, its verbs inflect for subject agreement, with the specific form of agreement markers determined by the verb's aspectual specification. The language features a full range of verb forms, including the perfective, imperfective, jussive, gerund, imperfective, and infinitive, and employs morphological markers to indicate causative and passive constructions (Demeke, 2017; Leslau, 1995).

This results in significant lexical variation, with orthographic words often encompassing multiple morphemes, including clitics and function words. For instance, the word ከየዛፉና (/käjjäzafunna/) encapsulates a complex structure composed of the preposition ከ- (/kä-/ “from”), a reduced distributive marker እየ- (/ijjä/ “each”), the noun ዛፍ (/zaf/ “tree”), the definite article -ኡ (/-u/ “the”), and the conjunction -ና (/-nna/ “and”). Clitics such as prepositions, conjunctions, and auxiliaries play critical syntactic roles but are not distinguished orthographically, making it difficult for automatic systems to segment and tag them accurately (Seyoum, Miyao, and Mekonnen, 2016). In such contexts, relying solely on syntactic position is insufficient for determining grammatical relationships; instead, models must integrate detailed morphological analysis to infer dependencies (Tsarfaty et al., 2010).

These issues are compounded by the fact that most high-performing parsers developed for languages like English do not generalize well to morphologically rich languages such as Arabic, Basque, and Greek due to their fundamentally different morphological structures (e.g., noun and verb inflection, free word order) (Nivre et al., 2007; Tsarfaty et al., 2010). As noted by Dehdari, Tounsi, and Genabith (2011), this morphological richness makes Amharic particularly challenging for parsing tasks. In addition, the absence of explicit markers for clitics in the script further complicates tasks like tokenization, POS tagging, and dependency parsing. Overall, these characteristics make Amharic a compelling case for employing state-of-the-art NLP toolkits that jointly consider morphological and syntactic information (e.g., Trankit). Despite the challenges of dependency parsing in morphologically rich languages (Seyoum, Miyao, and Mekonnen, 2018; Tsarfaty et al., 2010), transformer-based models show promise for such low-resource settings (Van Nguyen et al., 2021; Van Der Goot et al., 2020).

2.2 The Universal Dependencies Framework

The Universal Dependencies (UD) framework is built on the foundations of dependency grammar, offering a cross-linguistically consistent approach to syntactic and morphological annotations. This section first outlines the core concepts of dependency grammar, then explains how UD extends these principles to enable multilingual natural language processing and standardized treebank development.

2.2.1 Foundations in Dependency Grammar

Dependency grammar is a syntactic framework that models sentence structure by establishing direct binary relations between words, primarily between a head (a central element) and its dependents (modifiers, arguments, or complements) (Nivre, 2005). Instead of building structures around phrases like traditional phrase-structure grammar (e.g., noun phrases, verb phrases), dependency grammar eschews phrasal nodes and focuses on direct relationships between individual words in a sentence, organizing them based on their syntactic dependencies. (Tesnière, 2020; Hudson, 2007). This approach posits that every word in a sentence, except the root (typically the main verb), is linked to exactly one head, forming a tree structure without recursion or intermediate phrases (Kübler, McDonald, and Nivre, 2009). These dependencies are often labeled to specify syntactic or semantic roles, such as subject, object, or modifier, enabling precise representations of grammatical functions (De Marneffe and Nivre, 2019).

Dependency grammar's emphasis on word-to-word relationships offers computational advantages, particularly in natural language processing (NLP). Dependency parsers map sentences into dependency trees, efficiently supporting syntactic analysis-their primary function-while also enabling downstream applications such as machine translation and information extraction. They reduce complexity by avoiding phrase-level interaction (McDonald, Crammer, and Pereira, 2005). For example, in the sentence "The cat chased the mouse," the verb "chased" serves as the root, with "cat" as the subject (linked via an nsubj relation) and "mouse" as its direct object (dobj), while determiners ("the") attach as dependents to their nouns (Chen and Manning, 2014).

The framework's cross-linguistic applicability further enhances its utility. Languages with free word order (e.g., Latin) or minimal inflection (e.g., Chinese) can be analyzed effectively through dependency relations, which are less reliant on rigid constituent hierarchies (Buchholz and Marsi, 2006). This flexibility has made dependency grammar a cornerstone of modern computational linguistics, underpinning tools like Universal Dependencies (UD), a standard annotation system used for 150 languages (De Marneffe et al., 2021). Theoretical linguistics also employs dependency-based analyses to explore typological patterns and syntactic universals, arguing that such relations reflect cognitive processes in language production (Mel'cuk et al., 1988).

2.2.2 Principles and Structure of UD

A major initiative that has shaped the landscape of dependency grammar in recent years is the Universal Dependencies (UD) project, a comprehensive, community-driven effort aimed at creating a consistent framework for annotating grammatical relations across a diverse set of languages. The UD initiative was launched to address the challenges posed by the enormous variation in annotation schemes, which often hinder cross-linguistic research and comparison. Previous dependency treebanks for languages like Swedish, Danish, and English demonstrated significant discrepancies, with only about 40% of shared relations across the pairs, highlighting the difficulty of comparing dependency structures across languages (De Marneffe and Nivre, 2019; De Marneffe et al., 2021).

To overcome these challenges, UD introduces a standardized framework for morphosyntactic annotation that has been applied to more than 150 languages, producing over 200 treebanks. The aim is to enhance multilingual NLP applications, linguistic typology, and computational tasks such as parsing and machine learning, all while maintaining consistency across typologically diverse languages. For instance, the framework accommodates morphologically rich languages, pro-drop languages, and those with clitic doubling (De Marneffe et al., 2021; Nivre et al., 2020).

The UD framework standardizes three main layers of information in treebank annotation:

- Universal POS Tags (UPOS): A set of 17 part-of-speech categories applicable across all languages.
- Morphological Features: Fine-grained features (e.g., gender, number, case) that capture the rich internal structure of words.
- Dependency Relations: A standardized set of syntactic roles (e.g., nsubj, obj, obl) that define how words relate to each other.

The relations, while common across languages, exhibit typological variation in their realization. For example, in languages like English, grammatical relations are encoded using function words (e.g., determiners, auxiliaries) and word order, while in Finnish, case markers and inflections serve similar purposes without relying on explicit function words for encoding definiteness or tense (De Marneffe and Nivre, 2019). The table below presents a selection of Universal Dependency relations.

Clausal Argument Relations	
Relation	Description
nsubj	Nominal subject
obj	Direct object
iobj	Indirect object
ccomp	Clausal complement
xcomp	Open clausal complement
expl	Expletive
Nominal Modifier Relation	
Relation	Description
nmod	Nominal modifier
amod	Adjectival modifier
appos	Appositional Modifier
det	Determined
case	Prepositions, postpositions, and other case markers
nummod	Numeric modifier
compound	Compound noun modifier
Coordination and Linking Relations	
Relation	Description
aux	Auxiliary
cop	Copula (e.g., is, was)
mark	Marker (subordinating conjunction like that, because)
conj	Conjunct
cc	Coordinating conjunction
Adverbial and Discourse Relations	
Relation	Description
advmod	Adverbial modifier
discourse	Discourse element (e.g., well, oh)
Punctuation and Other	
Relation	Description
root	Root of the sentence
punct	Punctuation

Table 2.1: Selected Universal Dependency relations (De Marneffe et al., 2021)

UD's design philosophy prioritizes both cross-linguistic uniformity and language-specific flexibility. Annotations are grounded in a consistent set of syntactic and morphological relations, enabling comparison across languages while still allowing for language-specific extensions where needed. This balance has made UD the de facto standard for multilingual dependency treebanks and a cornerstone of major

NLP shared tasks, such as the CoNLL Universal Dependency parsing competitions (De Marneffe et al., 2021; Nivre et al., 2020).

The CoNLL-U format is the standardized data representation format used by the UD project. Each sentence in a UD treebank is represented as a list of token-level annotations in a plain-text, tab-separated format with 10-columns including information such as the token's ID, form (word), lemma (base form), Universal and language-specific part-of-speech tag, morphological features, head (syntactic governor), dependency relations, and optional metadata. Here's a breakdown of the 10 columns in a typical CoNLL-U file:

- ID: Token number in the sentence
- Form: The word form or punctuation symbol
- LEMMA: The lemma or stem of the word
- UPOS: Universal-part-of-speech tag
- XPOS: Language-specific part-of-speech tag
- FEATS: Morphological features (e.g., Gender=Fem|Number=Sing)
- HEAD: Head of the current token (by ID)
- DEPREL: Universal dependency relation to the HEAD
- DEPS: Enhanced dependency graph
- MISC: Miscellaneous annotations (e.g., alignment, named entity tags)

An example in CoNLL-U looks like as shown below in Table 2.2[1].

ID	FORM	LEMMA	UPOS	XPOS	FEATS	HEAD	DEPREL	DEPS	MISC
1	አስቴር	አስቴር	PROPN	PROPN	-	4	nsubj	-	Translit=
2	ወደ	ወደ	ADP	ADP	-	3	case	-	Translit=
3	ቤት	ቤት	NOUN	NOUN	-	4	obl:loc	-	Translit=
4-5	ሄደች	-	-	-	-	-	-	-	-
4	ሄድ	ሄድ	VERB	VERB	VF=Fin	0	root	-	Translit=
5	ኧች	እሱ	PRON	SUBJ	Gen=Fem	4	expl	-	LTranslit=
6	።	።	PUNCT	PUNCT	-	4	punct	-	Translit=

Table 2.2: CoNLL-U table for **"አስቴር ወደ ቤት ሄደች ። "** (English: Aster goes home.)

A significant feature of UD is its attention to content words, which are treated as the central elements of syntactic structures. Function words, such as determiners and prepositions, are connected to the content words they modify or depend on. This setup aligns with Tesniere's concept of the nucleus, where function words and content words form a unified syntactic unit (De Marneffe et al., 2021). This design

[1] NB: This treebank example omits parts of the FEATS and MISC columns for readability. Full annotations can be found in the original CoNLL-U file.

choice not only simplifies syntactic representation but also maximizes cross-lingual parallelism, enabling consistent annotation across typologically diverse languages. By standardizing around content-head structures, UD facilitates multilingual parsing, transfer learning, and comparative linguistic research, especially for low-resource languages.

In the Universal Dependencies (UD) framework, multi-word token (MWT) are used to represent such cases, where the surface form splits into multiple syntactic words (tokens), often due to affixes, clitics, contractions, or compound words (Nivre et al., 2020). This is common in Amharic, as many sentences in the official treebank exhibit such patterns.

In the CoNLL-U representation in Table 2.2 above, the tokens with IDs 4 (**ሄድ**) and 5 (**ኧች**) are analyzed as separate syntactic units – specifically, a verb stem and a pronoun suffix, respectively. However, these are also grouped together under a Multi-Word Token (MWT) spanning IDs 4–5 and represented as **"ሄደች"** in the MWT line. This MWT reflects the original surface form before morphological segmentation. In contrast, in the dependency tree visualization in Figure 2.1 below, this same construction is typically presented as a single boxed unit, preserving its syntactic unity while still acknowledging its internal morphological structure.

Let's examine a tree representation of the Amharic sentence **"አስቴር ወደ ቤት ሄደች ። "** (English: Aster goes home) as shown in Figure 2.1.

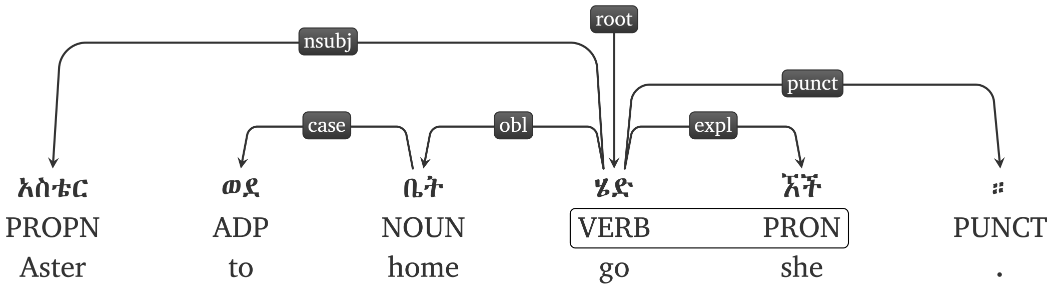


Figure 2.1: Syntactic dependency tree for **"አስቴር ወደ ቤት ሄደች ። "** (English: Aster goes home.)

The UD framework integrates insights from multiple existing standards, such as Stanford Dependencies and the Google universal part-of-speech tags, and draws from earlier efforts like Hamle DT (Rosa et al., 2014; Zeman et al., 2012) and the Universal Dependency Treebank Project (Nivre et al., 2020). By synthesizing these contributions, UD builds a unified annotation system that supports comparative linguistics and multilingual NLP tasks, such as cross-linguistic parsing and language model training.

The Universal Dependencies (UD) framework integrates insights from several existing standards, including Stanford Dependencies and Google's universal POS tags, while also building on earlier initiatives such as HamleDT and the Universal Dependency Treebank Project (Nivre et al., 2020).

UD's practical applications extend beyond theoretical linguistics, enabling a range of multilingual NLP technologies, from machine translation to automated question answering. The standardized dependency annotations facilitate more effective cross-linguistic learning, allowing for the development of systems that can handle low-resource languages. Furthermore, the ever-expanding UD corpus, the emergence of speech annotation in UD, now covering a broad set of languages, serves as a valuable resource for improving language processing tools and advancing cross-linguistic syntactic research (De Marneffe et al., 2021; Nivre et al., 2020).

The following tree representation illustrates the complexity of Amharic morphosyntax, using the sentence **“ዮሐንስ መጽሐፉን ለአስቴር መለሰላት ። ”** (English: Yohannes returns the book to Aster.) as shown in Figure 2.2.

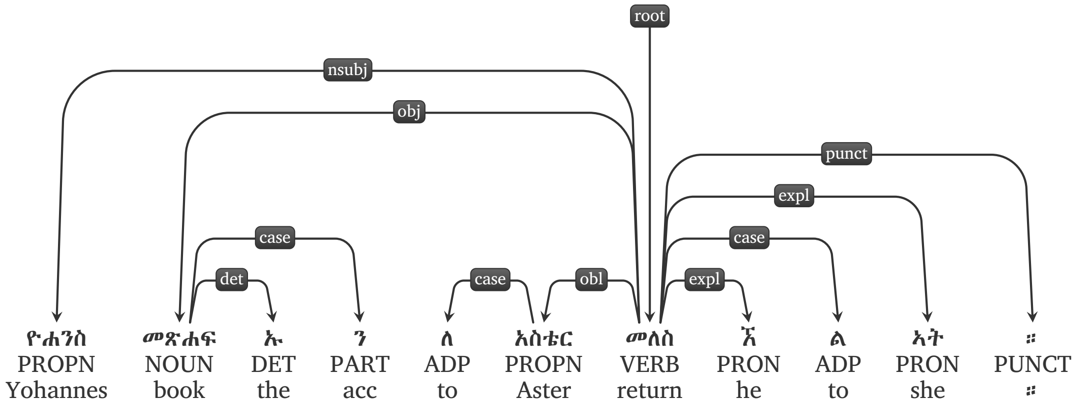


Figure 2.2: Syntactic dependency tree for **“ዮሐንስ መጽሐፉን ለአስቴር መለሰላት ። ”** (English: Yohannes returns the book to Aster.)

As seen in Figure 2.1 & 2.2 above, the fusion of functional morphemes into single orthographic tokens is especially challenging to parse, as it obscures syntactic boundaries and grammatical roles.

2.3 Dependency Parsing

Dependency parsing is a fundamental component of syntactic analysis in natural language processing (NLP), where the grammatical structure of a sentence is represented as a set of binary relations between words. Each relation connects a head word to a dependent, forming a tree or graph that reflects the underlying syntactic organization. This method is particularly useful for languages with flexible word order and rich morphology, as it emphasizes grammatical functions over word position. To contextualize dependency parsing within the broader scope of parsing in NLP, Section 2.3.1 outlines the main categories of parsing tasks; Section 2.3.2 discusses parsing approaches, with particular attention to dependency-based models.

2.3.1 Parsing Task Categories

Parsing tasks in NLP are typically classified into three main categories: syntactic parsing, semantic parsing, and morphological parsing (Jurafsky and Martin, 2021). Syntactic parsing focuses on the grammatical structure of a sentence, identifying the relationships between words. It includes constituency parsing, which organizes sentences into hierarchical parsing structures such as noun phrases (NP) and verb phrases (VP), and dependency parsing, which models grammatical relations between individual words rather than relying on phrase structure.

Semantic parsing goes beyond syntactic analysis to capture meaning, mapping sentences to logical forms or knowledge representations such as knowledge graphs. This is essential for applications like question answering and machine translation.

Morphological parsing, on the other hand, analyses the internal structure of words by breaking them down into morphemes: the smallest units of meaning. This task is particularly important for morphologically rich languages such as Amharic.

Among syntactic approaches, dependency parsing stands out as it models sentences as directed graphs, where words (dependents) are connected to head words via labeled grammatical relations (e.g., subject, object). Unlike the hierarchical grids of constituency parsing, dependency structures provide greater flexibility—especially for languages with free word order—since grammatical roles often take precedence over word position (Jurafsky and Martin, 2025, pp. 411-412).

Due to the language's rich and complex morphology – where information about tense, person, number, gender, and even syntactic roles is encoded within word forms – effective parsing must simultaneously resolve both the syntactic relationships between words and the internal morphological composition of individual words. For instance, verbs in Amharic often bundle subject, object, and tense information into a single complex form, requiring parsers to disentangle and represent these features accurately within a dependency structure. Thus, UD parsing for Amharic is inherently a hybrid process that bridges the boundary between syntactic structure and morphological decomposition.

2.3.2 Dependency Parsing Approaches

Parsing techniques in dependency parsing vary in how they construct syntactic structures, with major approaches including graph-based and transition-based . Transition-based parsers build dependency trees incrementally by performing local actions (such as shift, reduce, and arc creation) using a stack and buffer mechanism. These parsers are computationally efficient and fast, but they are susceptible to error propagation since early mistakes can affect the final structure (Kübler, McDonald, and Nivre, 2009).

Graph-based parsers, in contrast, treat dependency parsing as a global optimization task: they score all potential arcs between words and search for the highest-scoring complete tree, often using maximum spanning tree algorithms. This approach tends to achieve higher overall accuracy, particularly for languages with flexible word order, but typically at the cost of greater computational complexity (McDonald, Lerman, and Pereira, 2006; McDonald, Crammer, and Pereira, 2005).

For years, neural network-based methods have significantly advanced the field. Early models employed feed-forward networks and recurrent neural networks (RNNs) to predict parsing actions or dependency arcs. More recently, transformer-based architectures based on BERT and XLM-R, have become the standard for dependency parsing. These models generate deep contextual embeddings that capture complex syntactic and semantic information, enabling parsers to handle long-range dependencies and ambiguous constructions more effectively (Vaswani et al., 2017).

This evolution influences tool selection in Multilingual NLP. For instance, Trankit, a transformer-based pipeline, achieves high accuracy across diverse languages by combining contextual embeddings with efficient parsing strategies. The next chapter, Data and Methods, provides a detailed overview of Trankit’s framework.

2.4 Previous related Works

Despite growing interest in Amharic natural language processing (NLP), progress across core areas such as dependency parsing, UD treebank development, and tool evaluation remains fragmented. While some domains have seen notable advancements, others lag due to spare resources and methodological gaps. By synthesizing key studies, this section highlights how these efforts have shaped the development of critical resources like the Amharic Universal Dependencies (UD) Treebank- a breakthrough for enabling parser training and systematic analysis.

To enhance clarity and organization, the review of related literature is divided into two subsections: one focussing on previous work in Amharic NLP and POS tagging and the other on works related to treebank development and dependency parsing.

2.4.1 Amharic NLP and POS Tagging

Gambäck et al. (2009) pioneered early efforts in POS tagging but were constrained by the absence of large-scale annotated corpora. Their work demonstrated that rule-based taggers achieved 85% accuracy on a small dataset, while machine learning models (e.g., Hidden Markov Models) struggled with data sparsity. Recent advances in transformer-based models (e.g., BERT, GPT) have revolutionized NLP for high-resource languages, yet their application to Amharic remains underexplored.

Gebre (2010) achieved over 90% accuracy in POS tagging for Amharic, with conditional random fields (90.95%) outperforming support vector machines (90.43). These results marked a significant turning point, as previous efforts had not surpassed the 90% threshold. The improvement is attributed to a combination of factors, including cleaning and partial correction of the WIC corpus, linguistically informed feature selection (such as vowel patterns and root consonants characteristic of Semitic morphology), and careful parameter tuning in machine learning models.

In cross-linguistic comparison, although Arabic and Hebrew achieved higher accuracy due in part to smaller tagsets and higher-quality corpora, Gebre (2010)'s results—based on a 31-tag set and the largest training corpus—were promising, highlighting the potential for further enhancement with cleaner data and improved resources.

Complementing this line of research, Tonja et al. (2023) offers a broader survey of NLP developments across Ethiopian languages, including Amharic, Afaan Oromo, Tigrinya, and Wolaytta. Their findings further emphesize the limited visibility of local NLP research on global platforms, with much of the work still emerging from unpublished theses rather than peer-reviewed publications. This underscores the urgency of more comprehensive, scalable, and publicly accessible resources and tools, an issue with which the present project directly engages.

This effort aligns with broader calls for standardization in Amharic NLP. Tonja et al. (2023) emphasizes that inconsistent annotation practices and fragmented resources have stifled progress in Ethiopian languages. Their survey highlights that over 70% of Amharic NLP tools are developed using non-reproducible methodologies or proprietary datasets, limiting their utility for the broader research community. By contrast, the UD project's open-access ethos and community-driven guidelines offer a sustainable pathway for scalable resource development.

2.4.2 Treebank Development and Dependency Parsing for Amharic

Chief among these are the contributions by Seyoum, Miyao, and Mekonnen (2016), Seyoum, Miyao, and Mekonnen (2018), and Seyoum, Miyao, and Mekonnen (2020), whose foundational efforts in manual treebank development and empirical evaluation of dependency parsers have provided a baseline for subsequent research. Their work underscores the interplay between linguistic complexity and computational feasibility, yet also reveals unresolved issues: the scarcity of annotated data, cascading errors in preprocessing pipelines, and the need for theoretically grounded clitic-handling frameworks.

Seyoum, Miyao, and Mekonnen (2016) propose that the treebank annotation should consist of three tiers: part-of-speech (POS) tags, morphological features, and syntactic relations. In addition to segmentation challenges, Seyoum, Miyao, and Mekonnen (2016) underscore the broader linguistic complexities of Amharic. A central focus of Seyoum, Miyao, and Mekonnen (2016) is the segmentation of clitics – grammatical elements that are often fused with content words in Amharic and present a major challenge for accurate tokenization and syntactic analysis. Seyoum, Miyao, and Mekonnen (2016)'s process includes developing annotation guidelines, resolving morphological and syntactic ambiguities (e.g., distinguishing active and passive forms), and achieving high inter-annotator agreement (>95%). They adopt an iterative workflow that combines semi-automatic POS tagging with manual corrections, ensuring a balance between scalability and accuracy.

Their methodology reflects a linguistically grounded approach tailored to the specific characteristics of morphologically rich languages. The authors argue that existing resources and tools fail to adequately handle the fusion of content and function words, prompting them to develop detailed guidelines for the manual segmentation of clitics. Recognizing the influence of phonological and orthographic variations, Seyoum, Miyao, and Mekonnen (2016) emphasizes the necessity of linguistic expertise in the annotation process.

A summary of the primary challenges and proposed solutions-including some contributions by the present author-is provided in Table 2.3 below.

Challenges	Implications	Potential Solutions
Morphological Complexity	Data sparsity, parsing inaccuracy	Hybrid models, morphological segmentation
Script Ambiguities	Tokenization errors, orthographic variance	Standardization, gemination markers
Cliticization/Syntactic Words	Phrasal tokenization	Rule-based segmentation, syntactic annotation
Compound Word Variability	Inconsistent tokenization	Orthographic reform, corpus normalization

Table 2.3: Key challenges in Amharic NLP and proposed solutions

Seyoum, Miyao, and Mekonnen (2018) introduced the first Universal Dependencies (UD) Treebank for Amharic, consisting of 1,074 manually annotated sentences.[2]

[2] https://github.com/Binyamephrem/Amharic-treebank

Given Amharic's morphological richness and the scarcity of existing tools, the labor-intensive nature of manual annotation is justified.

They emphasized that, due to the lack of standardization in the Amharic writing system, "some people tend to write in phonemic form (the abstract form or what one intends to say), while others write in phonetic form (what is actually uttered)" (Seyoum, Miyao, and Mekonnen, 2018). As a result, the authors deliberately excluded data from social media and similar sources, where such inconsistencies in writing are more prevalent. Based on the Universal Dependencies (UD) framework, they proposed language-specific part-of-speech (POS) tags, morphological features, and syntactic dependency types tailored for Amharic.

By addressing these issues, the authors lay the groundwork for effective UD annotation in Amharic. Their insight that "a word in Amharic often leaves boundaries of lexical or syntactic units unclear" reflects the central linguistic hurdle they aim to overcome. The strengths of this work lie in its linguistic precision, especially in identifying phonological shifts and distinguishing between affixes and clitics. However, the reliance on manual annotation presents limitations for scalability, and the theoretical ambiguity surrounding clitics versus affixes remains unresolved. Moreover, questions remain about the representativeness of the corpus across genres.

Despite these limitations, the treebank fills a critical gap in Amharic NLP, enabling the development of tools such as POS taggers and syntactic parsers. Its UD-aligned annotations provide a robust foundation for future research on low-resource languages. Suggested future directions include the automation of clitic segmentation, expansion of annotated corpora, and theoretical refinement of morphosyntactic categories.

Seyoum, Miyao, and Mekonnen (2020) shift focus to dependency parsing, evaluating four neural network-based parsers – UDPipe, jPTDP, UUParser, and the Turku Neural Parser – on the manually annotated UD treebank. Using standard evaluation metrics such as Labeled Attachment Score (LAS), Unlabeled Attachment Score (UAS), Morphological LAS (MLAS), and Bilexical LAS (BLEX), they conduct a ten-fold cross-validation to account for data scarcity. Seyoum, Miyao, and Mekonnen (2020) reported that the UDPipe-based model achieved a performance of 62.08% Unlabeled Attachment Score (UAS) and 55.32% Labeled Attachment Score (LAS) on raw Amharic text.

The study reveals that neural models can effectively learn syntactic dependencies from small datasets, particularly when pre-trained embeddings mitigate out-of-vocabulary issues. Character-level architectures such as that of jPTDP have also been shown to be beneficial for handling morphologically complex tokens (Seyoum, Miyao, and Mekonnen, 2020). However, the pipeline nature of many parsers introduces cascading errors – segmentation inaccuracies, for instance, can significantly degrade downstream performance.

They concluded that despite the small size of the treebank, neural dependency parsers can perform well in Amharic, provided that the training data capture sufficient syntactic diversity. They recommend further development in three key areas: improving segmentation quality, integrating morphological tagging into joint models, and mitigating pipeline-related error propagation. Their work underscores both the promise and the persistent challenges of NLP for underresourced, morphologically complex languages.

Degu and Gebeyehu (2022) based their initial dataset on the Amharic treebank annotated by Seyoum, Miyao, and Mekonnen (2018), which contains 1,074 sentences, plus their own non-UD-compliant 500 sentences [3] . Three classifiers were developed and trained: a transition action classifier, a relation label classifier, and a POS tag classifier. The dataset was preprocessed and randomly split into 70% for training and 30% for testing.

Degu and Gebeyehu (2022) proposed an LSTM-based dependency parsing system for Amharic and evaluated it using 30% of the dataset, reporting an accuracy of 91.54% (UAS) and 86% (LAS). However, the reported results lack clarity regarding the evaluation setup. It is likely that these scores were obtained using sentences in which the orthographic tokens were left unsegmented – that is, not analysed further into linguistically meaningful units – as observed in the non-UD-compliant version of the Treebank. This simplification reduces parsing complexity and may inflate performance metrics. Their methodology included the creation of a 500-sentence treebank; however, this corpus lacks compliance with Universal Dependencies (UD) standards, omitting critical annotation layers such as #sent_id, #text, #translit, #translit, LEMMA, XPOS (language-specific POS tags), FEATS (morphological features), and MISC (miscellaneous metadata). Those omissions limit the treebank's utility for cross-linguistic benchmarking and likely contribute to parsing inaccuracies.

For instance, when parsing the sentence **"አበበ እሮጠ ። "** ("Abebe ran"), their system misidentifies the proper noun **"አበበ"** (PRON) as the root and erroneously labels the word **"እሮጠ"** (VERB) as a nominal subject case marker (SUBJ), a nonstandard label under UD guidelines [4]. This results in a dependency structure where the verb - the syntactic head - is incorrectly linked to the subject. The misanalysis reflects structural limitations in the parser's ability to accurately handle basic Amharic sentence constructions.

Yeshambel, Mothe, and Assabie (2024) made strides in this direction by releasing a comprehensive collection of Amharic resources for information retrieval, including raw text, stem-and root-based corpora, a stopword list, stem/root lexicons, and WordNet-like resources. They also developed word embedding trained on both raw and morphologically segmented corpora, carefully accounting for Amharic's

[3] `https://github.com/mizgithub/Amharic-Treebank-dataset`

[4] `https://mizgithub-amharic-dependency-parserapp-index-459fri.streamlit.app`

rich morphology. Their findings suggest that root-based normalization improves morphological generalization; however, word-based models still outperform root-based ones in retrieval tasks (58.7% vs 56.2% F1-score). This highlights the need for hybrid approaches that integrate morphological decomposition with syntactic context. These resources, which were evaluated both experimentally and by linguists, address a critical gap in Amharic NLP and are now publicly available to support future research.

2.4.3 Persistent Gaps and Future Directions

Despite progress, three critical gaps persist:

- Data Scarcity: The Amharic UD Treebank (1,074 sentences) (Seyoum, Miyao, and Mekonnen, 2018) remains too small for training transformer-based models. For comparison, the English UD Treebank contains over 16,000 sentences.
- Theoretical Ambiguity: The clitic/affix distinction in Amharic morphology lacks consensus, leading to inconsistent annotation. Seyoum, Miyao, and Mekonnen (2018) manually resolved such cases, but automated solutions are needed for scalability.
- Tool Fragmentation: Existing tools (e.g., tokenizers, POS taggers) are often isolated and not easily interoperable, with limited compatibility across frameworks like SpaCy, Stanza, or Hugging Face.

By synthesizing these insights, this thesis identifies persistent gaps and proposes directions to advance Amharic NLP toward parity with high-resource languages; see Table 2.4.

Study	Contribution	Limitations
Gambäck et al. (2009)	POS tagged corpus	High dependency on manual intervention
Gebre (2010)	Advanced Amharic POS tagging	Quality of the corpus
Seyoum, Miyao, and Mekonnen (2016)	Creation of a foundational resource	Annotation Methodology Concerns
Seyoum, Miyao, and Mekonnen (2018)	First UD Treebank for Amharic	Smal corpus size (1,096 sentences)
Seyoum, Miyao, and Mekonnen (2020)	Comparative Evaluation of Neural Parsers for Amharic	Dependency on Pre-trained Resources
Degu and Gebeyehu (2022)	Develop parser	Non-UD annotations, parsing errors
Tonja et al. (2023)	Pan-Ethiopian NLP survey	Highlights gaps but offers no new tools
Yeshambel, Mothe, and Assabie (2024)	Resources tailored for information retrieval	Underperform compared to word-based models in retrieval tasks

Table 2.4: Some of the recent advances and unmet needs in Amharic NLP

The reviewed works collectively underscore the unique challenges of Amharic and the gradual progress made in addressing them. Foundational resources such as the UD Treebank and recent corpora have enabled the development of initial parsers. However, issues of scalability, standardization, and integration with modern NLP frameworks continue to pose significant barriers.

This project aims to bridge this gap by expanding the treebank and reinforcing the importance of annotation quality and training state-of-the-art parsing architecture like Trankit. As part of the efforts to expand the existing treebank, this project involves creating a new treebank through annotation from scratch and upgrading a portion of Degu and Gebeyehu (2022)'s non-UD-compliant treebank to full UD compliance – including the addition of critical fields such as #sent_id, #text, #translit, lemma, MWTs, xpos, feats, translit, misc, and consistent relation labelling. These enhancements contribute to improved syntactic parsing accuracy.

3 Data and Methods

Building on the foundation laid in Chapter 2, which reviewed the challenges and limitations in existing Amharic dependency parsing work, this chapter outlines the datasets, preprocessing strategies, and experimental methodologies employed to evaluate transformer-based parsing models. Particularly, it addresses the limitations in previous works by refining and augmenting existing datasets to meet Universal Dependencies (UD) standards and by enhancing a state-of-the-art multilingual NLP toolkit. Through additional annotation and enhancement of the Degu and Gebeyehu (2022) treebank[1], the corpus was expanded by 330 sentences and 3273 tokens, resulting in improved linguistic coverage and data quality for model evaluation.

3.1 Data Source and Preprocessing

The primary dataset used for this study is the Amharic UD Treebank developed by Seyoum, Miyao, and Mekonnen (2018). The treebank consists of 1074 manually annotated sentences sourced from news articles and literary texts, adding conversational language and offering a balanced domain representation[2]. It includes detailed morphological segmentation, POS tagging, and syntactic dependencies in accordance with UD version 2.15. However, its limited size and genre coverage restrict its overall representativeness for broader linguistic analysis. To address these gaps, a two-stage enhancement was implemented: treebank expansion and validation using the UD tools.

3.1.1 Treebank Expansion

To enhance both the quantity and syntactic diversity of the training data, an additional 330 sentences – comprising 3273 tokens – were semi-manually annotated and integrated into the existing treebank. These sentences were drawn from a balanced collection of contemporary Amharic texts, including news articles and public domain literature. As discussed in the previous chapter, a treebank of 330 sentences was annotated from scratch, with approximately 180 of these upgraded to

[1] `https://github.com/mizgithub/Amharic-Treebank-dataset`

[2] `https://github.com/UniversalDependencies/UD_Amharic-ATT`

a fully UD-compliant state from a partially annotated, non-UD-compliant treebank by Degu and Gebeyehu (2022)[3]. The newly annotated sentences were sourced from contemporary literary works and books on history and politics.

All annotations in this thesis were checked for consistency with Seyoum, Miyao, and Mekonnen (2018). The final 330-sentence treebank is currently available on the following GitHub repository [4], and later will be publicly released and incorporated into the official UD treebank repository[5].

3.1.2 UD Compliance Conversion

To ensure consistency and interoperability with the UD framework, the extended Treebank underwent a rigorous refinement process, resulting in a total of 1402 UD-annotated sentences. It involved a combination of automated (e.g., filling in missing lemmas) and extensive manual (e.g., MWTs, FEATS, and MISC) processing.

- Lemmatization: Standardization of word roots was carried out using the Ethiopic Lemma Lexicon. This step mitigated orthographic variation and improved consistency in morphological analysis.
- The Feats fields were populated with verb-specific morphological information-such as VerbForm=Fin and Aspect=Perf-using standardized templets (Seyoum et al., 2016) that were manually reviewed and adjusted.
- Clitic Segmentation: Manual Rule-based methods were applied to segment fused morphemes. For example, the compound word ከየዛፉና (”and from each tree”) was segmented into its constituent morphemes: ከ+የ+ዛፍ+ኡ+ና.
- Dependency Label Alignment: Non-standard labels from the original corpus (e.g., SUBJC) were systematically mapped to UD equivalents (e.g., nsubj). Inconsistencies were manually resolved on the basis of native-speaker judgment.

3.2 Methodological Framework

3.2.1 Trankit Framework and Custom Pipeline for Amharic

Trankit is a pipeline-based natural language processing (NLP) toolkit that performs core tasks – tokenization, sentence splitting, POS tagging, morphological feature tagging, lemmatization, named entity recognition (NER), and dependency parsing(Van Nguyen et al., 2021). Trankit uses a transformer-based approach for dependency parsing. It builds on neural network models (e.g., transformer-based architecture like XLM-RoBERTa) that process input tokens incrementally through a sequence of actions (e.g., shift, reduce) to construct a dependency tree.

[3] https://github.com/mizgithub/Amharic-Treebank-dataset

[4] https://github.com/Jembda/MLT-Thesis-Trankit

[5] https://github.com/UniversalDependencies/UD_Amharic-ADT/tree/master

Built on XLM-RoBERTa, a multilingual transformer-based model trained on over 100 languages, Trankit provides 90 pretrained pipelines covering 56 languages. However, Amharic is not included among the languages with ready-to-use pretrained pipelines.

Despite this, Amharic is one of Trankit's trainable languages, meaning that custom pipelines can be built from scratch using Universal Dependencies (UD) formatted treebank. Trankit supports this through the TPipeline class for training and Pipeline for deployment, organizing customized pipelines into four categories depending on which NLP tasks are included. In the present work, a customized-mwt pipeline was developed for Amharic, which includes the following components:

- Joint token and sentence splitter
- Multi-word token expander
- Joint POS, morphological tagger, and dependency parser
- Lemmatizer

To train the Amharic pipeline (for sentence segmenter and tokenizer), raw text files were paired with semi-manually annotated .conllu files derived from the expanded and refined treebank (Section 3.1). Each module was trained separately using Trankit's Application Programming Interface (API), with development sets used to monitor performance and avoid overfitting. Notably, the training data were carefully segmented and verified for consistency in morphology, dependency relations, and tokenization boundaries.

Once all components were trained, the pipeline was verified using Trankit's verify_customized_pipeline utility, confirming the successful integration of all model components. After verification, the Amharic pipeline was loaded and used like any other pretrained language model supported by Trankit.

Trankit was chosen over MaChAmp due to its holistic approach to morphosyntactic analysis, where POS tagging, morphological analysis, and dependency parsing are jointly performed in a unified third module of the pipeline. This joint modeling is particularly beneficial for Amharic, a morphologically rich language, as it allows the parser to make more informed decisions by leveraging interdependent linguistic features. Additionally, Trankit offers native support for training on new languages using multilingual embeddings like XLM-RoBERTa, making it well-suited for under-resourced languages like Amharic.

The present work represents the first custom-trained Trankit pipeline for Amharic, enabling an transformer-based dependency parsing system especially tailored to the linguistic complexities of the language[6].

[6] https://github.com/nlp-uoregon/trankit

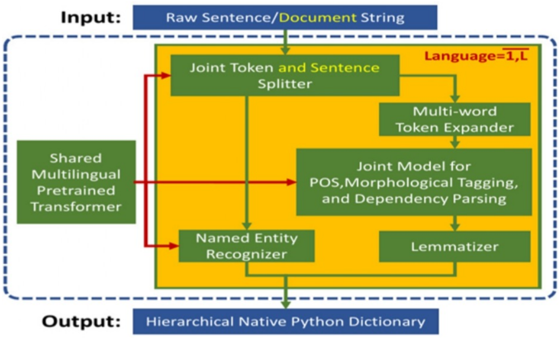

Input:
Raw Sentence/Document String
Language=1,L
Joint Token and Sentence Splitter
Multi-word Token Expander
Shared Multilingual Pretrained Transformer
Joint Model for POS,Morphological Tagging, and Dependency Parsing
Named Entity Recognizer
Lemmatizer
Output:
Hierarchical Native Python Dictionary

Figure 3.1: Trankit's Architecture (Van Nguyen et al., 2021)

3.2.2 Experimental Setup

To ensure consistent evaluation and comparability across parsers, the existing Amharic UD Treebank (Seyoum, Miyao, and Mekonnen, 2018) is split into training, evaluation and test sets using the standard 80/10/10 split—widely adopted in dependency parsing, NLP, and machine learning.

Similarly, to assess the impact of data cleaning on model performance, the expanded treebank –including the newly annotated 330 sentences –is also split into training, validation, and test sets following the same 80/10/10 ratio.

Dataset	Training Set	Development Set	Test Set
UD Treebank	859	107	108
Expanded dataset	1041	130	131

Table 3.1: Dataset splits for training, development, and evaluation

The splits were carefully designed to maintain a balanced distribution of sentence lengths, genres, and syntactic complexity across the different sets, minimizing sampling bias and ensuring that the evaluation results fairly reflect generalization capabilities (De Marneffe et al., 2021).

In addition to evaluating transformer-based parsing models trained via Trankit, the results of this thesis were compared with baseline performance figures reported by Seyoum, Miyao, and Mekonnen (2020) on raw text data, which used UDPipe and Turku parsing toolkits. UDPipe representing a transition-based architecture based on feature-driven methods rather than deep contextual embeddings, allowing for a clear contrast between conventional parsing approaches and transformer-based models.

The use of UDPipe as a baseline enables a quantitative assessment of the improvements gained through modern transformer architectures, particularly in handling the morphological richness and syntactic complexity of Amharic.

3.2.3 Evaluation Metrics

Model performance was evaluated across multiple linguistic layers, reflecting the sequential nature of the parsing pipeline. The evaluation metrics were categorized into three primary areas: tokenization and sentence splitting, morphosyntactic performance, and dependency parsing performance.

I Tokenization and Sentence Splitting: At the initial stages of the pipeline, accurate tokenization and sentence boundary detection are crucial for downstream tasks. The following metrics were used:

- Token F1 Score: Measures the harmonic mean of the precision and recall for token boundaries.
- Sentence F1 Score: Measures the precision and recall of sentence segmentation.

II Morphosyntactic Performance:

The following metrics were evaluated:

- Words: In Trankit, the words metric evaluates tokenization accuracy by measuring how well the system-generated tokens align with the gold-standard tokens in the annotated dataset. This reflects the Trankit-based model's ability to correctly identify word boundaries, a crucial step in morphologically rich languages like Amharic, where incorrect tokenization can cascade into error in POS tagging and dependency parsing (Van Nguyen et al., 2021). A high score in this metric indicates precise word segmentation consistent with Universal Dependecies guidelines.
- UPOS Accuracy: measures the correctness of universal part-of-speech tagging.
- XPOS Accuracy: measures the correctness of language-specific parts-of-speech tagging (where available)
- UFeats Accuracy: measures the correctness of morphological features prediction.
- AllTags Accuracy measures the combined correctness across UPOS, XPOS, and UFeats annotations.
- Lemma F1 Score: measures the precision and recall of lemmatization, accounting for orthographic normalization and root extraction.

III Dependency Parsing Performance: The final stage assesses the model's ability to predict syntactic structures:

- Labeled Attachment Score (LAS): Measures the percentage of words that are assigned both the correct syntactic head and the correct dependency label. It is considered the gold standard for evaluating parsing quality.
- Unlabeled Attachment Score (UAS): Measures the percentage of words correctly attached to their syntactic heads, regardless of the specific dependency label. This metric isolates attachment quality from label classification.
- Content Word Labeled Attachment Score (CLAS): similar to LAS but restricted to content words (e.g., nouns, verbs, adjectives), providing a focused view of syntactic heads.
- Morphology-Aware Labeled Attachment Score (MLAS): extends LAS by also requiring correct morphological features on nodes and their syntactic heads.
- Bi-lexical Dependency Score (BLEX) combines LAS with correct lemmatization of head to evaluate both syntactic and lexical predictions.

Together, these evaluation metrics provide a comprehensive understanding of model performance across tokenization, morphosyntactic annotation, and syntactic parsing. This multi-level evaluation enables a robust comparison between neural network-based parsing approaches and modern transformer-based models.

3.3 Ethical Considerations

The development of language processing tools, particularly for low-resource languages such as Amharic, raises important ethical considerations related to dataset composition, representativeness, and potential biases.

Ensuring transparency in model design and data usage is essential for building trust in NLP tools. If dependency parsers are used in downstream applications like machine translation or speech interfaces, undocumented decisions can hinder reproducibility and raise concerns about reliability. To address this, it is important to clearly document the data sources, annotation guidelines, training settings, and known limitations. This promotes accountability and enables validate, compare, or build upon the work

A notable concern in this study is bias. The refined corpus predominantly are drawn from formal domains, such as government publications, religious texts, literary works, and grammar manuals. As a result, the language style and syntactic structures represented in the dataset may not fully capture the diversity of Amharic as it is used across different social contexts, such as informal speech, regional dialects, or social media communication. This overrepresentation of formal registers could lead to parsing models that are less effective when applied to colloquial or contemporary language varieties.

Efforts were made during the annotation refinement phase to ensure internal consistency and linguistic accuracy; however, future work should prioritize broadening the treebank to include more informal, user-generated content such as online forums, social media posts, and conversational transcripts. Expanding the dataset in this way would contribute to the development of more robust, inclusive parsing models capable of handling a wide range of Amharic linguistic phenomena.

Additionally, ethical annotation practices were maintained throughout the refinement process, including reliance on native speaker adjustment to resolve ambiguities and complexities, thus minimizing annotation errors that could propagate bias or misrepresentation of syntactic structures.

3.4 Limitations

Despite the contributions made in expanding and refining the Amharic dependency treebanks, several limitations were encountered that constrain the generalizability and performance of the resulting parsing models.

3.4.1 Data Scarcity

Despite the incorporation of 330 additional semi-manually annotated treebanks, the overall dataset remains small compared to those available for high-resource languages like English. This limited data size constrains the toolkit's capacity to learn infrequent syntactic patterns, generalize across genres, and maintain consistent performance across diverse text types. In particular, the lack of informal or conventional Amharic within the treebank restricts the customized model's applicability in the real-world scenarios outside formal or literal contexts.

Consequently, the Trankit's performance in highly dependent on both the size and the quality of the available training data. Unlike models for high-resource languages, which benefit from pretraining on vast amounts of diverse text, Trankit-trained Amharic parsers must rely on relatively small and domain-specific treebank. Furthermore, while Trankit leverages XLM-RoBERTa-a powerful multilingual toolkit-it is not specifically optimized for the morphosyntactic characteristics of Amharic, such as rich verb morphology, and extensive cliticization. This mismatch may hinder the parser's ability to accurately capture fine-grained syntactic and morphological distinctions essential for high-quality dependency parsing in Amharic.

3.4.2 Toolkit Constraint

While Trankit provides a valuable framework for training complete pipelines for dependency parsing, it is not without limitations. One major limitation is the lack of modular usability—while some components (e.g., tagging or parsing) can be trained individually, they cannot be used independently of the full pipeline. For example, there is no straightforward way to test the parser or tagger on pre-tokenized text, nor to evaluate only the tokenizer or sentence splitter in isolation. If an error arises in one module, such as the tokenization phase, it causes the entire pipeline training to fail, impeding iterative development. Moreover, the toolkit suffers from limited documentation and minimal support from developers. For example, an issue raised regarding a specific error message on Trankit's official discussion forum has remained unanswered for an extended period, highlighting a lack of active community or maintainer engagement.

Additionally, Trankit demonstrates reliance on non-core columns like MISC; during the experiments, omitting the MISC column – even when it seemed unnecessary – resulted in a dramatic drop in parsing performance. The toolkit is also heavily dependent on metadata fields within the input files, and removing or misformatting these fields can cause the training process to fail entirely. These constraints limit Trankit's robustness.

3.5 Practical Experience with Trankit and Alternatives

During the training phases, several practical challenges and considerations emerged when working with Trankit, particularly compared to the alternative toolkit MaChAmp. Although Trankit was ultimately chosen due to its holistic approach to morphosyntactic approach and strong empirical performance on joint lexical and syntactic tasks (Van Nguyen et al., 2021), the path to effective implementation proved more difficult than initially anticipated. The original plan was to train the Amharic treebanks using Trankit's customizable pipeline functionality, which integrates tokenization, POS tagging, lemmatization, and dependency parsing in a unified framework.

However, repeated failures during training –particularly within the tokenization component –halted progress, as Trankit's architecture does not allow individual modules to be tested or debugged in isolation. One such issue was posted on Trankit's GitHub discussion page but, as of this writing, remains unanswered – highlighting the toolkit's limited community support and responsiveness.

As a result, I temporarily turned to MaChAmp (Van Der Goot et al., 2020), a modular multitask NLP framework designed to accommodate multiple input formats and custom task configurations. MaChaAmp's flexible architecture and transparent training procedures provided a more manageable development experience and served as a practical fallback during this period.

Eventually, after multiple failed attempts, I revisited Trankit and discovered two critical adjustments that enable successful model training. First, I filled in missing lemma fields in the treebank, which were previously left empty and had contributed to unexpected crashes. Second, I modified the training configuration by setting ‘char_level’ = True, a crucial parameter for heading Ge’ez script languages such as Amharic.

Setting char_level_ = True is important for processing Amharic, a morphologically rich language written in Ge’ez script. This setting allows us to analyse text at the character level, which helps capture sub-word patterns such as affixes, clitics, and root forms, especially in morphologically complex languages (Cotterell et al., 2018). It also improves generalization to rare forms – an advantage for low-resource languages like Amharic where a large annotated treebank is lacking (Ponti et al., 2019).

In addition, I simplified the pipeline category by removing the –ner suffix (i.e., changing customized-mwt-ner to customized-mwt), thereby excluding the named entity recognition component, which was not a focus of the experiments in this thesis. Ultimately, many issues had to be resolved through source code inspection and extensive trial and error. With the modifications in place, Trankit successfully began training. Consequently, I abandoned the alternative MaChAmp toolkit and restarted the full training and evaluation pipeline using Trankit.

Trankit was trained as the main pipeline for three key reasons: (1) its superior performance across NLP tasks on development data relevant to this thesis; (2) its relatively compact and unified architecture, which simplified downstream deployment; and (3) the feasibility of adapting the pipeline to new datasets once its structure and undocumented errors, idiosyncrasies in column usage (e.g., reliance on the MISC field), were eventually resolved through careful tuning, code review, and iterative testing.

4 Results and Discussion

The chapter presents the results and accompanying discussion, structured into three major evaluation categories: (i) tokenization and sentence splitting, (ii) morphosyntactic performance, and (iii) dependency parsing. It begins with a comparative evaluation of Trankit against the results reported by Seyoum, Miyao, and Mekonnen (2020) for UDPipe and the Turku Parsers, offering a clear benchmarking of progress in Amharic natural language processing. The comparison highlights Trankit's strength across a range of linguistic tasks, with notable improvements in morphosyntactic analysis and in partial dependency parsing.

Following this, the chapter presents the results of experiments conducted on both the original Amharic UD Treebank (Seyoum, Miyao, and Mekonnen, 2018) and the newly expanded Treebank. Particular attention is given to evaluating the relative impact of data cleaning versus data volume on the performance of the Trankit parser. To this end, experiments were conducted using three versions of the dataset: (1) the original Amharic UD Treebank (V2.15), (2) the expanded dataset prior to refinement, and (3) the cleaned, expanded dataset. Particular attention is given to the effects of dataset cleaning on overall parsing accuracy. The structured presentation allows for both a tool-based performance comparison and an assessment of the impact of data quality enhancement in improving NLP outcomes.

4.1 Dataset Overview

Training and evaluation were conducted using two versions of the dataset: the original UD Treebank (Seyoum, Miyao, and Mekonnen, 2018) and an expanded version, which added 330 annotated sentences. Table 4.1 summarizes the dataset splits.

Dataset	Training Set	Development Set	Test Set
UD Treebank (2018)	859	107	108
Expanded Treebank	1041	130	131

Table 4.1: Dataset splits for training, development, and evaluation

The expanded Treebank aimed to improve coverage, though it initially introduced performance variations, which were later addressed through cleaning, and provided a more diverse set of examples, particularly for underrepresented syntactic and morphological phenomena.

4.2 Comparison of the Trankit-based model against UDPipe-based & Turku-based models

This section presents a comparative analysis of the performance of the Trankit-based model on the original Amharic UD Treebank (Seyoum, Miyao, and Mekonnen, 2018) (V 2.15) against the UDPipe-based and Turku-based models results reported by Seyoum et al. (2020). It is important to note, however, that the results reported by Seyoum, Miyao, and Mekonnen (2018) are based on cross-validation using nearly the entire dataset for training, whereas the present study evaluates performance using an 80/10/10 train/dev/test split, thus relying on substantially less training data. This highlights the strength of the present approach, which achieves comparable, when not significantly better, results by leveraging Trankit's transformer-based architecture. However, direct comparisons between the models should be interpreted with caution, as they are based on different evaluation strategies.

The evaluation spans multiple core linguistic tasks, including tokenization, sentence segmentation, part-of-speech tagging, lemmatization, morphological features, and syntactic dependency parsing. The comparison aims to assess the extent of progress in Amharic language processing and identify areas of strength and limitation across toolkits.

4.2.1 Tokenization and Sentence Splitting

In this category, UDPipe achieved the highest performance in token segmentation with a perfect score of 100.00%, followed by Turku (99.70%), and Trankit(99.09%). For sentence segmentation, both UDPipe and Turku showed equal strength (98.62%), outperforming Trankit (97.22%). While Trankit slightly lags behind, its results in Table 4.2 still demonstrate a high degree of quality.

Metrics	Trankit	UDPipe	Turku
Token f1 score	99.09	100.00	99.70
Sentence f1 score	97.22	98.62	98.62

Table 4.2: Trankit vs UDPipe & Turku (Seyoum et al., 2020)

4.2.2 Morphosyntactic Performance

The Trankit-based model clearly outperforms both UDPipe-based and Turku-based models in most morphosyntactic metrics. It achieved the highest score in word segmentation (86.48%), UPOS tagging (81.62%), XPOS tagging (80.89%), UFeats (79.44%), AllTags (75.71%), and lemmatization (84.93%). In comparison, UDPipe and Turku hovered around 80.00% for word-level and lemma accuracy, with notably lower POS tagging and morphological feature extraction, as can be seen in Table 4.3 below.

Metrics	Trankit	UDPipe	Turku
Words	86.48	80.23	80.07
UPOS	81.62	75.94	77.14
XPOS	80.89	75.38	76.91
UFeats	79.44	73.38	74.24
AllTags	75.71	72.23	74.66
Lemmas	84.93	80.23	80.07

Table 4.3: Trankit vs UDPipe & Turku (Seyoum et al., 2020)

4.2.3 Dependency Parsing

When it comes to syntactic parsing, Trankit slightly outperforms both UDpipe and Turku in UAS with 63.59%. However, Turku shows superior performance in parsing metrics:

- LAS: Turku (55.63%) > Trankit (55.52%) > UDPipe (55.32%)
- CLAS, MLAS, and BLEX: Turku leads across all three with 49.96%, 46.06%, and 49.96%, respectively, suggesting more precise and semantically aware parsing structures.

Metrics	Trankit	UDPipe	Turku
UAS	63.59	62.08	61.60
LAS	55.52	55.32	55.63
CLAS	48.91	49.33	49.96
MLAS	40.65	42.74	46.06
BLEX	48.10	49.33	49.96

Table 4.4: Trankit vs UDPipe & Turku (Seyoum, Miyao, and Mekonnen, 2020)

4.2.4 Nature of Dataset Cleaning and its Effects

The cleaning process involved several key strategies aimed at improving the consistency and reliability of the dataset for paring tasks: The cleaning process involved two main strategies:

a) Correction of transcription inconsistencies in the MISC fields, which previously affected token alignment, morphological analysis, and lexical features.

b) Fill in missing metadata, such as language-specific features and morphological attributes, to enhance the completeness of the annotation and support more accurate parsing.

c) Correction of inaccurate dependency labels and root assignments, which are critical for the syntactic integrity of the tree structures and directly influence parsing accuracy.

d) Segmentation of previously unsegmented clitics, a common issue in Amharic due to its complex morphology. Proper segmentation ensures that syntactic relationships are represented more transparently and in accordance with the UD guidelines.

These reinforcements contribute to noticeable improvement in model performance, demonstrating that data quality-particularly syntactic and morphological consistency-plays a critical role in parsing outcomes, sometimes even more than data volume.

After cleaning the expanded dataset, performance generally improved across tokenization, sentence splitting, and dependency parsing. While there were some slight declines in morphosyntactic performance, these changes are relatively minor, suggesting that the cleaning process helped refine the dataset while maintaining a strong performance in the core tasks. This demonstrates the model's sensitivity to consistent annotation. Importantly, sentence splitting retained perfect accuracy throughout, indicating its robustness across dataset versions. The improvements in UAS, CLAS, and BLEX particularly highlight the positive impact of cleaning on dependency parsing, with better handling of syntactic and lexical dependencies.

4.3 The Impact of Data Quality vs. Quantity on Parsing Performance

This section examines the relative impact of data cleaning versus data volume on the performance of the Trankit Parser. To do this, experiments were conducted on three datasets:

- The original Amharic UD Treebank 2018 (V.2.15) reported in section 4.2
- The expanded Treebank Pre-cleaning, and
- The expanded Treebank Post-cleaning

The goal is to understand whether increased data quantity alone is sufficient to improve parsing accuracy or if systematic cleaning and normalization are necessary to fully realize the benefits of additional data.

Metric	F1 Score (2018 Treebank)	F1 Score (Pre-cleaning)	F1 Score (Post-cleaning)
Tokens	99.09	95.01	98.28
Sentences	97.22	100.00	100.00
Words	86.48	81.42	85.05
UPOS	81.62	76.16	79.93
XPOS	80.89	74.95	78.93
UFeats	79.44	73.41	77.25
AllTags	75.71	69.85	73.47
Lemmas	84.93	79.89	82.45
UAS	63.59	58.60	64.23
LAS	55.52	50.58	55.58
CLAS	48.91	44.94	50.13
MLAS	40.65	37.00	41.37
BLEX	48.10	44.44	48.71

Table 4.5: Results Pre-cleaning and Post-cleaning

The comparison between the original and expanded treebanks accounts for the fact that cleaning was performed after the treebank was made UD-compliant, with only small and systematic changes introduced. This cleaning was prompted by a substantial decline in performance across the evaluation metrics observed with the pre-cleaned expanded treebank.

4.3.1 Pre-Cleaning Results

Following the expansion of the UD Treebank with an additional 330 annotated sentences, an initial evaluation was conducted. The model trained on the expanded dataset exhibited noticeable changes in performance compared to the original UD Treebank. The results are presented according to three categories: Tokenization and Sentence Splitting, Morphosyntactic Performance, and Dependency Parsing Performance.

I Tokenization and Sentence Splitting: While sentence segmentation slightly improved to 100% accuracy, tokenization performance dropped by 4 points, suggesting that the additional sentences may have introduced inconsistencies in token-boundaries.

II Morphosyntactic Performance: Morphosyntactic performance declined consistently across all metrics, with an average drop of about 5–6 F1 points. This performance degradation indicates increased noise and annotation variability in the expanded dataset.

III Dependency Parsing Performance: Dependency Parsing Accuracy was notably impacted, with a decline of approximately 4-5 points across all parsing metrics.

This reflects the sensitivity of syntactic parsers to annotation consistency and quality.

The initial evaluation revealed that while sentence segmentation was robustly improved, overall tokenization, morphosyntactic annotation, and syntactic parsing performance suffered after expanding the dataset. These findings strongly indicated that the newly added 330 sentences introduced inconsistencies and errors in the data, underscoring the need for a comprehensive cleaning phase to improve dataset quality before further model development.

4.3.2 Post-Cleaning Results

After clearing the expanded dataset, the performance was re-evaluated to examine the impact of cleaning. The results of this evaluation are presented in Table 4.5 above, comparing the results obtained on the cleaned expanded Treebank to the results obtained on the UD Treebank (Seyoum, Miyao, and Mekonnen, 2018).

Data cleaning led to notable advancements across multiple evaluation metrics when model performance on Expanded Treebank (Pre-cleaning), and Expanded Treebank (Post-cleaning) is compared. Tokenization accuracy improved by 3.27%, and sentence segmentation remains a perfect 100% score. Lexical and morphological tagging saw consistent gains: word recognition (+3.65%), UPOS (+3.77%), XPOS (+3.98%), UFeats (+3.84%), and combined AllTags (+3.62%). Lemmatization improved by 2.56%.

Syntactic parsing showed substantial improvements: unlabelled attachment score (UAS), increased by 5.63%, labelled attachment score (LAS) by 5.00%, and both CLAS and MLAS improved by 5.13% and 4.37% respectively. BLEX, which evaluates syntactic and lexical accuracy, rose by 4.27%. These results confirm that syntactic data cleaning significantly boosts parsing accuracy.

Besides, the results can be examined across the three datasets – Original Treebank (2018), Expanded Treebank (Pre-cleaning), and Expanded Treebank (Post-cleaning) – as follows:

I Tokenization and Sentence Splitting: Both tokenization and sentence splitting performed excellently, with a minor improvement in token segmentation after cleaning. The sentence splitting remained perfect, demonstrating no change in its high accuracy post-cleaning.

II Morphosyntactic Performance: Despite some minor declines in morphosyntactic performance, the changes are relatively small. The introduction of new data likely added complexity and diversity, which may have led to these slight declines in POS tagging, morphological feature recognition, and lemmatization.

III Dependency Parsing Performance: Dependency parsing showed minor improvements in most metrics after cleaning, with UAS showing a slight increase and

CLAS, MLAS, and BLEX seeing consistent gains. These improvements suggest that the cleaning process helped the model better handle syntactic and lexical dependencies.

4.4 Discussion

The experimental results offer compelling insights into the challenges and improvements associated with expanding and refining Amharic UD Treebank. The comparisons across the original UD Treebank (2018), the expanded (Pre-cleaning) dataset, and the cleaned version of the expanded dataset allow for several important observations. These comparisons allow us to assess how each tool adapts to different data conditions and highlight the critical role of resource preparation in achieving robust linguistic analysis.

4.4.1 Trankit's Performance and Advancements

Trankit demonstrates significant improvements over transition-based tools such as UDPipe and Turku. It consistently outperforms both baselines in core linguistic tasks, including tokenization, sentence segmentation, POS tagging (UPOS/XPOS), morphological feature extraction (UFeats), and lemmatization. These advantages are evident across both the original UD Treebank (Seyoum, Miyao, and Mekonnen, 2020) and the extended Treebank (Post-cleaning).

In terms of dependency parsing, Trankit performs competitively on metrics such as the Unlabeled Attachment Score (UAS), Labeled Attachment Score (LAS), and the enhanced Labeled Attachment Score (CLAS). However, it trails slightly in Morphology-Aware Labeled Attachment Score (MLAS) and BiLexical Dependencies (BLEX), which evaluate a model's ability to integrate morphological analysis with syntactic structure. These lower scores suggest that, while Trankit excels in lexical analysis, transition-based toolkits like Turku may still have a marginal advantage in morphosyntactic integrations.

The impact of dataset quality is also clear. Training in the expanded Treebank (Post-cleaning) leads to measurable gains in performance, in sentence segmentation, which reaches 100% accuracy, and especially in syntactic metrics such as CLAS and UAS. This reinforces the importance of data cleaning and consistency in training transformer-based models effectively. Overall, Trankit proves to be a highly competitive toolkit for processing plain text in low-resource languages. It shows strong lexical and tagging accuracy and remains competitive in syntactic analysis. Although Turku still maintains a slight advantage.

4.4.2 Impact of Dataset Expansion

The expansion of the dataset introduced more linguistic variety, but it also brought in inconsistencies and errors that affected the model's ability to generalize. This is especially evident in the initial drop in performance across almost all morphosyntactic and syntactic metrics on the expanded Treebank. For instance, the F1 score for Universal POS tags (UPOS) dropped from 81.62% (original) to 76.16% (expanded before cleaning), and UAS dropped from 63.59% to 58.60%. This decline points to issues such as inconsistent annotation practices, increased structured diversity, or incorrect feature labeling within the newly added data.

Such decline is common in low-resource settings, where additional data does not always yield improved performance – particularly when the annotation quality is not aligned with established UD guidelines (Nivre et al., 2020).

4.4.3 Error Analysis and Annotation Challenges

Manual inspection of the original Treebank (Seyoum, Miyao, and Mekonnen, 2018) revealed morphosyntactic syntactic errors, such as gender mismatches in pronominal annotations, empty lemmas, and inconsistent tokenization of enclitics and fused forms. These problems were partially corrected during the cleaning process. However, they highlight the need for refined annotation guidelines and inter-annotator agreement protocols for future expansion efforts.

Despite improvements, minor performance declines in morphosyntactic metrics (e.g., UPOS and XPOS) compared to the model's performance on the Treebank by Seyoum, Miyao, and Mekonnen (2018) point to persistent annotation noise. Errors such as the misannotation of grammatical gender can have cascading effects on multiple parsing layers. These challenges highlight the need for stronger validation processes and possibly a shift toward collaborative annotation in future developments.

5 Conclusion

This thesis set out to improve the parsing performance by training a transformer-based model using Trankit. It further explores the effects of expanding and cleaning the Amharic UD Treebank.

The experiments reveal that Trankit, a state-of-the-art toolkit, not only excels in token-level tasks but also performs competitively in dependency parsing, achieving strong results in metrics such as UAS, LAS, and CLAS. However, it shows slightly lower performance morphosyntactic integration metrics (MLAS and BLEX), where transition-based toolkits like Turku maintain a slight advantage.

Notably, the Trankit-based model delivers comparable parsing performance despite being trained on just 80% of the available Treebank, highlighting its efficiency in low-resource settings. In contrast, the results reported by Seyoum, Miyao, and Mekonnen (2020) were obtained through cross-validation on nearly the full Treebank, whereas this study adopts an 80/10/10 train/test/dev split, thereby relying on considerably less training data. These findings suggest that while Trankit-based models are highly effective for surface-level linguistic analysis, further enhancements may be necessary to fully capture the integration of morphological and syntactic structures.

These performance gains are further amplified when the Trankit toolkit is trained on a cleaned and systematically expanded version of the Amharic Treebank. However, it is worth noting that the Turku-based model may still retain a marginal advantage in certain aspects of morphosyntactic integration, suggesting room for further refinement in Trankit's handling of complex grammatical structures.

A key insight from this research is the crucial importance of data quality. Cleaning the extended dataset resulted in clear performance gains across nearly all evaluation metrics, particularly in sentence segmentation (which reached 100% accuracy) and syntactic parsing.

The experimental results presented in chapter 4 demonstrate that while dataset expansion can introduce valuable linguistic diversity, it must be paired with rigorous quality control to avoid degrading model performance. Clearing the dataset significantly recovered accuracy in tokenization and dependency parsing, confirming the central hypothesis that annotation consistency is as important as data volume.

5.1 Key contributions

- Demonstrated Trankit's comparable or superior performance –depending on the specific tasks – over transition-based toolkits on the original 2018 Amharic Treebank, while using less training data.
- Introduced and validated a systematically expanded and cleaned version of Amharic UD Treebank, adding approximately 330 annotated sentences and correcting inconsistencies in tokenization, sentence segmentation, Multi-word-token (MWT) handling, morphosyntactic features, lemmatization, and syntactic dependencies.
- Demonstrated that even modest annotation noise can significantly impact model performance, especially in morphosyntactic and syntactic parsing tasks-emphasizing the importance of quality over quantity in low-resource language settings.
- Quantified the impact of dataset cleaning on performance, showing measurable improvements in tokenization, dependency parsing (UAS, LAS, CLAS, MLAS, BLEX), and partial recovery of morphosyntactic accuracy.
- Developed Trankit-based model for Amharic based on the cleaned and expanded UD Treebank, covering core tasks such as tokenization, POS tagging, morphological analysis, lemmatization, and dependency parsing. The resulting model is currently available via a public GitHub repository [1], and a formal request has been submitted for its integration into the official Trankit website to enhance accessibility and visibility. This model is suitable for a range of downstream applications, including machine translation, sentiment analysis, and information extraction.
- Provided a thorough experience report on the challenges of applying a transformer-based toolkit to an under-resourced language, offering insights that may inform similar efforts in other languages or domains.

5.2 Future Work

Several directions for future research emerge from this study:

- Further Expansion of the Treebank : Incorporate more diverse text genres, including informal and conversational Amharic, to improve robustness and real-world applicability.

[1] https://github.com/Jembda/MLT-Thesis-Trankit

- Data Quality Improvements: Establish rigorous guidelines and validation procedures to ensure higher annotation consistency and accuracy in morphosyntactic and syntactic labels, which directly impact model performance.
- Cross-lingual Transfer Learning: Explore leveraging related Semitic languages or other low-resource languages (e.g., Tigrinya, Oromo) through multilingual or transfer learning approaches to enhance parsing accuracy.
- Comparative Evaluation with Other Toolkits: Run the enhanced Treebank through multiple NLP toolkits (MaChAmp, UDpipe, Turku) and systematically compare performances to validate generalizability and toolkit-specific strength.

In conclusion, this study confirms that modern state-of-the-art toolkits like Trankit – when trained on a clean, curated Treebank – can not only match but in many cases surpass neural tools. While challenges remain in fully integrating morphosyntactic structures, the results reaffirm the growing promise of transformer-based models in expanding the linguistic reach of NLP.

Bibliography

Buchholz, Sabine and Erwin Marsi (2006). “CoNLL-X shared task on multilingual dependency parsing.” In: Proceedings of the tenth conference on computational natural language learning (CoNLL-X), pp. 149–164.

Central Statistical Agency of Ethiopia (2007). Population and Housing Census Report. Addis Ababa, Ethiopia: Central Statistical Agency (CSA).

Chen, Danqi and Christopher D Manning (2014). “A fast and accurate dependency parser using neural networks.” In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 740–750.

Cotterell, Ryan, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Luo Que (2018). “Character-based neural morphological tagging for richly inflected languages.” In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). Association for Computational Linguistics, pp. 1481–1491.

De Marneffe, Marie-Catherine, Christopher D. Manning, Joakim Nivre, and Daniel Zeman (July 2021). “Universal Dependencies.” In: Computational Linguistics 47.2, pp. 255–308. ISSN: 0891-2017. DOI: 10.1162/coli_a_00402. eprint: https://direct.mit.edu/coli/article-pdf/47/2/255/1938138/coli_a_00402.pdf. URL: https://doi.org/10.1162/coli_a_00402.

De Marneffe, Marie-Catherine and Joakim Nivre (2019). “Dependency grammar.” In: Annual Review of Linguistics 5.1, pp. 197–218.

Degu, Mesfin Zewdie and Wondimagegn Bekele Gebeyehu (2022). “Development of Dependency Parser for Amharic Sentences.” In: Proceedings of the 2nd Deep Learning Indaba-X Ethiopia Conference 2021. Conference held January 27–29, 2022. Addis Ababa, Ethiopia.

Dehdari, Jon, Lamia Tounsi, and Josef van Genabith (2011). “Morphological features for parsing morphologically-rich languages: A case of Arabic.” In: Proceedings of the Second Workshop on Statistical Parsing of Morphologically Rich Languages, pp. 12–21.

Demeke, Girma Awgichew (2017). Amharic Grammar. Addis Ababa: Addis Ababa University Press.

Gambäck, Björn, Fredrik Olsson, Atelach Alemu Argaw, and Lars Asker (2009). “Methods for Amharic part-of-speech tagging.” In: First Workshop on Language Technologies for African Languages, March 2009, Athens, Greece.

Gebre, Binyam Gebrekidan (2010). "Part of speech tagging for Amharic." PhD thesis. University of Wolverhampton Wolverhampton.

Habash, Nizar Y (2010). Introduction to Arabic natural language processing. Morgan & Claypool Publishers.

Hudson, Richard A. (2007). Language Networks: The New Word Grammar. Oxford University Press UK.

Jurafsky, Daniel and James H Martin (2021). Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.

Jurafsky, Daniel and James H. Martin (2025). Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models. 3rd. Online manuscript released January 12, 2025. URL: https://web.stanford.edu/~jurafsky/slp3/.

Kübler, Sandra, Ryan McDonald, and Joakim Nivre (2009). Dependency Parsing. Vol. 2. Synthesis Lectures on Human Language Technologies 1. Morgan & Claypool, pp. 1–127. DOI: 10.2200/S00169ED1V01Y200901HLT002.

Leslau, Wolf (1995). Reference Grammar of Amharic. Wiesbaden: Otto Harrassowitz. ISBN: 3-447-03372-X.

Manning, Christopher and Hinrich Schutze (1999). Foundations of statistical natural language processing. MIT press.

McDonald, Ryan, Koby Crammer, and Fernando Pereira (2005). "Online large-margin training of dependency parsers." In: Proceedings of the 43rd annual meeting of the association for computational linguistics (ACL'05), pp. 91–98.

McDonald, Ryan, Kevin Lerman, and Fernando Pereira (2006). "Multilingual dependency analysis with a two-stage discriminative parser." In: Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X), pp. 216–220.

Mel'cuk, Igor Aleksandrovic et al. (1988). Dependency syntax: theory and practice. SUNY press.

Nivre, Joakim (2005). Dependency grammar and dependency parsing. Tech. rep. MSI report 05133. Växjö University.

Nivre, Joakim, Marie-Catherine De Marneffe, Filip Ginter, Jan Hajič, Christopher D Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman (2020). "Universal Dependencies v2: An evergrowing multilingual treebank collection." In: arXiv preprint arXiv:2004.10643.

Nivre, Joakim, Johan Hall, Jens Nilsson, Atanas Chanev, Güls□en Eryigit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi (2007). "MaltParser: A language-independent system for data-driven dependency parsing." In: Natural Language Engineering 13.2, pp. 95–135.

Ponti, Edoardo Maria, Ivan Vulić, Goran Glavaš, Roi Reichart, and Anna Korhonen (2019). "Modeling language variation and universals: A survey on typological linguistics for natural language processing." In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, pp. 4525–4549.

Rosa, Rudolf, Jan Masek, David Marecek, Martin Popel, Daniel Zeman, and Zdenek Zabokrtskỳ (2014). "HamleDT 2.0: Thirty Dependency Treebanks Stanfordized." In: LREC, pp. 2334–2341.

Seyoum, Binyam Ephrem, Yusuke Miyao, and Baye Yimam Mekonnen (2016). "Morpho-syntactically Annotated Amharic Treebank." In: CLiF, pp. 48–57.

— (2018). "Universal dependencies for Amharic." In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).

— (2020). "Comparing Neural Network Parsers for a Less-resourced and Morphologically-rich Language: Amharic Dependency Parser." In: Proceedings of the first workshop on Resources for African Indigenous Languages, pp. 25–30.

Tesnière, Lucien (2020). Éléments de syntaxe structurale. Paris: Klincksieck.

Tonja, Atnafu Lambebo, Tadesse Destaw Belay, Israel Abebe Azime, Abinew Ali Ayele, Moges Ahmed Mehamed, Olga Kolesnikova, and Seid Muhie Yimam (2023). "Natural language processing in ethiopian languages: Current state, challenges, and opportunities." In: arXiv preprint arXiv:2303.14406.

Tsarfaty, Reut, Djamé Seddah, Yoav Goldberg, Sandra Kübler, Yannick Versley, Marie Candito, Jennifer Foster, Ines Rehbein, and Lamia Tounsi (2010). "Statistical parsing of morphologically rich languages (spmrl) what, how and whither." In: Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages, pp. 1–12.

Van Der Goot, Rob, Ahmet Üstün, Alan Ramponi, Ibrahim Sharaf, and Barbara Plank (2020). "Massive choice, ample tasks (MaChAmp): A toolkit for multi-task learning in NLP." In: arXiv preprint arXiv:2005.14672.

Van Nguyen, Minh, Viet Dac Lai, Amir Pouran Ben Veyseh, and Thien Huu Nguyen (2021). "Trankit: A light-weight transformer-based toolkit for multilingual natural language processing." In: arXiv preprint arXiv:2101.03289.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, □ukasz Kaiser, and Illia Polosukhin (2017). "Attention is all you need." In: Advances in neural information processing systems 30.

World Bank (2024). World Development Indicators – Ethiopia. Washington, D.C.: World Bank.

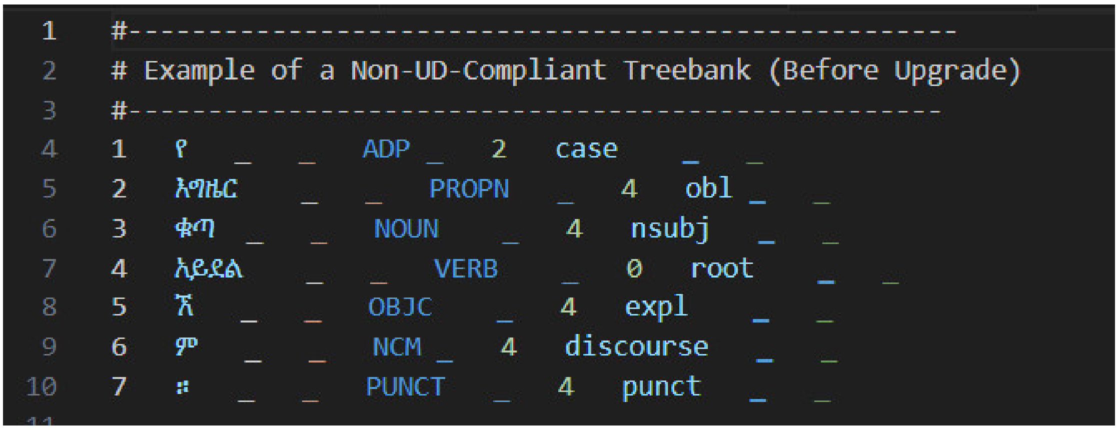
Yeshambel, Tilahun, Josiane Mothe, and Yaregal Assabie (2024). "Construction of Amharic information retrieval resources and corpora." In: Language Resources and Evaluation 58.4, pp. 1157–1185.

Zeman, Daniel, David Marecek, Martin Popel, Loganathan Ramasamy, Jan Stepánek, Zdenek Zabokrtskỳ, and Jan Hajic (2012). "Hamledt: To parse or not to parse?" In: LREC, pp. 2735–2741.

Appendix

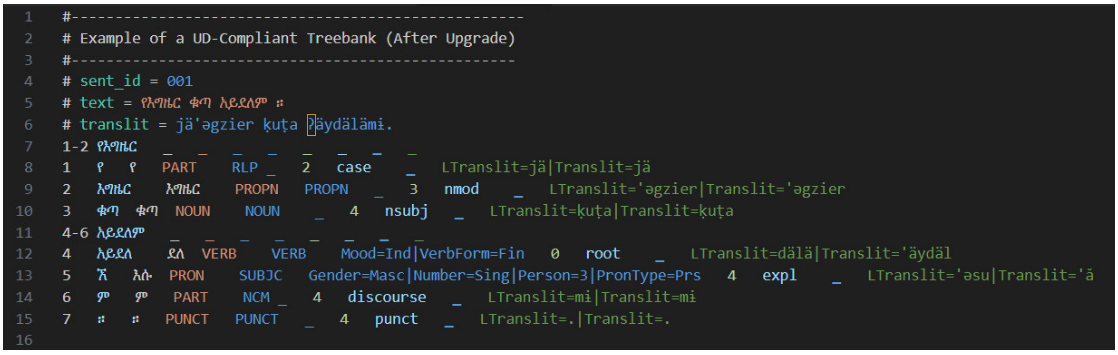
A Dataset sample

The following are data samples in CONLL-U format, drawn from stages of the Amharic treebank expansion:



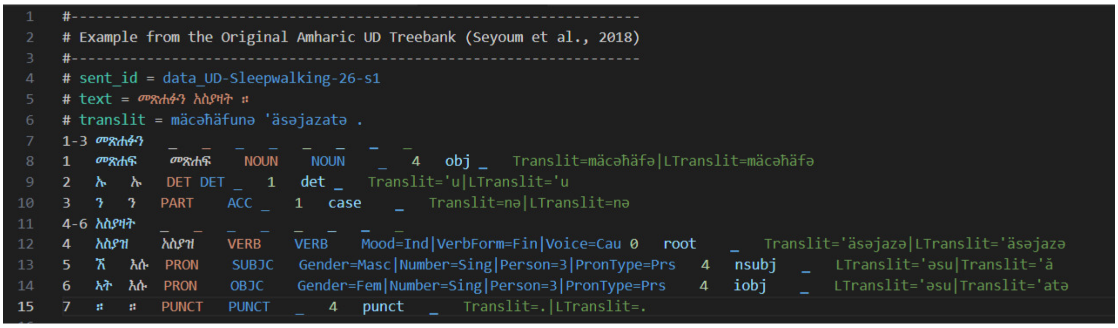
```
#----------------------------------------------------
# Example of a Non-UD-Compliant Treebank (Before Upgrade)
#---------------------------------------------------
1   የ   _   _   ADP _   2   case    _   _
2   እግዜር   _   _   PROPN   _   4   obl _   _
3   ቁጣ  _   _   NOUN    _   4   nsubj   _   _
4   አይደል   _   _   VERB    _   0   root    _   _
5   ኸ   _   _   OBJC    _   4   expl    _   _
6   ም   _   _   NCM _   4   discourse   _   _
7   ።   _   _   PUNCT   _   4   punct   _   _
```

Figure A.1: Dataset sample 1



```
#----------------------------------------------------
# Example of a UD-Compliant Treebank (After Upgrade)
#---------------------------------------------------
# sent_id = 001
# text = የእግዜር ቁጣ አይደለም ።
# translit = jä'əgzier ḳuṭa ʔäydälämɨ.
1-2 የእግዜር   _   _   _   _   _   _   _   _
1   የ   የ   PART    RLP _   2   case    _   LTranslit=jä|Translit=jä
2   እግዜር    እግዜር    PROPN   PROPN   _   3   nmod    _   LTranslit='əgzier|Translit='əgzier
3   ቁጣ  ቁጣ  NOUN    NOUN    _   4   nsubj   _   LTranslit=ḳuṭa|Translit=ḳuṭa
4-6 አይደለም   _   _   _   _   _   _   _   _
4   አይደለ    ደለ  VERB    VERB    Mood=Ind|VerbForm=Fin   0   root    _   LTranslit=dälä|Translit='äydäl
5   ኧ   እሱ  PRON    SUBJC   Gender=Masc|Number=Sing|Person=3|PronType=Prs   4   expl    _   LTranslit='əsu|Translit='ă
6   ም   ም   PART    NCM _   4   discourse   _   LTranslit=mɨ|Translit=mɨ
7   ።   ።   PUNCT   PUNCT   _   4   punct   _   LTranslit=.|Translit=.
```

Figure A.2: Dataset sample 2



```
#-----------------------------------------------------------------
# Example from the Original Amharic UD Treebank (Seyoum et al., 2018)
#-----------------------------------------------------------------
# sent_id = data_UD-Sleepwalking-26-s1
# text = መጽሐፉን አስያዛት ።
# translit = mäcəħäfunə 'äsəjazatə .
1-3 መጽሐፉን   _   _   _   _   _   _   _   _
1   መጽሐፍ    መጽሐፍ    NOUN    NOUN    _   4   obj _   Translit=mäcəħäfə|LTranslit=mäcəħäfə
2   ኡ   ኡ   DET DET _   1   det _   Translit='u|LTranslit='u
3   ን   ን   PART    ACC _   1   case    _   Translit=nə|LTranslit=nə
4-6 አስያዛት   _   _   _   _   _   _   _   _
4   አስያዝ    አስያዝ    VERB    VERB    Mood=Ind|VerbForm=Fin|Voice=Cau 0   root    _   Translit='äsəjazə|LTranslit='äsəjazə
5   ኧ   እሱ  PRON    SUBJC   Gender=Masc|Number=Sing|Person=3|PronType=Prs   4   nsubj   _   LTranslit='əsu|Translit='ă
6   አት  እሱ  PRON    OBJC    Gender=Fem|Number=Sing|Person=3|PronType=Prs    4   iobj    _   LTranslit='əsu|Translit='atə
7   ።   ።   PUNCT   PUNCT   _   4   punct   _   Translit=.|LTranslit=.
```

Figure A.3: Dataset sample 3