ON THE ANNOTATION DIFFERENCES/STYLES BETWEEN TURKISH TREEBANKS

BÜŞRA YAŞAR

ÖZYEĞİN UNIVERSITY

JANUARY, 2025

ON THE ANNOTATION DIFFERENCES/STYLES BETWEEN TURKISH TREEBANKS

by
Büşra Yaşar

Thesis
Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science

in
Computer Science

Advisor: Prof. Olcay Taner Yıldız

Graduate School of Science and Engineering
Özyeğin University
İstanbul

January, 2025

ON THE ANNOTATION DIFFERENCES/STYLES BETWEEN TURKISH TREEBANKS

Approved by:

Prof. Olcay Taner Yıldız, Advisor
Department of Artificial Intelligence and Data Engineering
Özyeğin University

Prof. Arzucan Özgür
Department of Computer Engineering
Boğaziçi University

Asst. Prof. İlknur Karadeniz
Department of Artificial Intelligence and Data Engineering
Özyeğin University

Approval Date: November 12, 2024

To my beloved mother, my entire family, my esteemed advisor, and my dear friends, whose unwavering support and encouragement guided me throughout this thesis journey.

DECLARATION OF ORIGINALITY

I hereby declare that I am the sole author of this thesis and that this is the true copy of my thesis, including the final revisions, approved by my thesis committee. All data and information have been obtained, produced, and presented in accordance with the rules of research ethics and principles of academic honesty. As required by these rules, to the best of my knowledge I have acknowledged ideas, thoughts, and any copyrighted material in accordance with the standard referencing rules. I certify that any part of this thesis has not been submitted for a degree or diploma in another educational institution.

Büşra Yaşar

ABSTRACT

The aim of this thesis is to detect differences between treebanks for Turkish treebanks accepted by Universal Dependencies [1], to observe the improvement by making changes in word-level operations such as stem selection and feature labeling, and to present a study that creates an annotation standard for Turkish treebanks. In this study, experimental operations were implemented to reveal the annotation differences between treebanks. These operations are concatenating the splitted words currently available in Turkish treebanks and performing their morphological analysis to understand a grammatical structures of treebanks, determining differences by analyzing word root choices in these treebanks, and determining feature annotation differences between treebanks by making feature comparisons at word by word. We hope the observation on differences between treebanks will reduce the annotation differences between Turkish treebanks, let to obtain more stable and grammatically more accurate roots and features in Turkish treebanks and will be a base study and resource for creating treebank annotation standards for Turkish.

Keywords: Turkish, treebank, CoNLL-U Format used for Universal Dependencies (CONLLU), annotated sentence, natural language processing, dependency parsing

ÖZET

Bu tezin amacı, Evrensel Bağımlılıklar tarafından kabul edilen Türkçe ağaç bankaları için ağaç bankaları arasındaki farkları tespit etmek, kök seçimi ve özellik etiketlemesi gibi kelime düzeyindeki işlemlerde değişiklikler yaparak iyileştirmeyi gözlemlemek ve Türkçe ağaç bankaları için bir işaretleme standardı oluşturan bir çalışma sunmaktır. Bu tez çalışmasında, ağaç bankaları arasındaki işaretleme farklılıklarını ortaya çıkarmak için deneysel işlemler uygulanmıştır. Bu deneysel işlemler, şu anda bu ağaç bankalarında bulunan bölünmüş kelimeleri birleştirerek ve ağaç bankalarının dilbilgisi yapılarını anlamak için morfolojik analizlerini yapmak, ağaç bankalarındaki kelime kök seçimlerini analiz ederek farklılıkları belirlemek ve kelime kelime özellik karşılaştırmaları yaparak veri kümeleri arasındaki özellik işaretleme farklılıklarını belirlemektir. Veri kümeleri arasındaki farklılıkların gözlemlenmesiyle Türkçe ağaç bankaları arasındaki işaretleme farklılıklarının azaltılacağını, Türkçe ağaç bankalarında daha kararlı ve dilbilgisi açısından daha doğru kökler ve özellikler işaretleneceğini ve Türkçe için ağaç bankası işaretleme standartları oluşturmak için temel bir kaynak oluşturulacağını umuyoruz.

Anahtar Kelimeler: Türkçe, ağaç bankası, CONLLU, açıklamalı cümle, doğal dil işleme, bağımlılık ayrıştırma

ACKNOWLEDGEMENTS

I would like to sincerely thank my advisor, Prof. Olcay Taner Yıldız, for his guidance, support, and patience throughout this research. His advice, encouragement, and feedback were invaluable to me.

I am also deeply grateful to the members of my thesis committee, Prof. Arzucan Özgür and Asst. Prof. İlknur Karadeniz, for their time, expertise, and constructive criticism, which greatly improved the quality of this study.

Lastly, I would like to express my heartfelt gratitude to my family, especially my mother, Gülen Yaşar, whose love and support I continue to feel deeply, even after her passing. Her unconditional love, encouragement, and sacrifices have been the foundation of this achievement. I would also like to thank my dear friends for their unwavering support throughout this journey.

This thesis is dedicated to all those who believed in me and provided the strength and motivation to persevere.

Thank you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS AND ABBREVIATIONS

SD	Stanford Dependencies
UD	Universal Dependencies
NLP	Natural Language Processing
AI	Artificial Intelligence
HamleDT	Harmonized Multi-Language Dependency Treebanks
POS	Part-of-Speech
LAS	Labeled Attachment Score
UAS	Unlabeled Attachment Score
LS	Label Score
ATIS	Air Travel Information System
IG	nflectional groups
BRAT	Brat Rapid Annotation Tool.
UPOS	Universal Part-of-Speech
MISC	Miscellaneous
XPOS	Extended Part of Speech
TNC-UD	Turkish National Corpus Universal Dependency
METU	Middle East Technical University
DA	Locative Suffix
SYM	Symbol
NUMMOD	Numerical Modifier

LF	Line Feed characters
BOUN	Boğaziçi University
NOM	Nominative
CCONJ	Coordinating Conjunction
GEN	Genitive
ITU	Istanbul Technical University
PL	Plural
PROPN	Proper Noun
DB	Derivational Boundary
IMST	ITU-METU-Sabancı Treebank
UDT	Universal Dependency Treebank
DET	Determiner
AOR	Aorist Tense
PROP	Proposition
AUX	Auxiliary
NUM	Number
GB	Grammar Book
CONLLU	CoNLL-U Format used for Universal Dependencies
GPU	Graphics Processing Unit
PUNCT	Punctuation
INTJ	Interjection
LEMMA	Base Form of a Word

PRON	Pronoun
CDT	Contemporary Dictionary of Turkish
SG	Singular
ID	Identifier
PWN	Princeton WordNet
OBJ	Object
ADP	Adposition
PNON	Possessive Non-Agreement
ADV	Adverb
WP	Wh-Pronoun
DI	Past Tense Suffix
SCONJ	Subordinating Conjunction
UK	United Kingdom
ACC	Accusative
LOC	Locative
HEAD	Head of a Phrase
TR	Turkish
DET	Determiner
USA	United States of America
XCOMP	Open Clausal Complement
VN	Verb-Noun
ATIS	Air Travel Information System

COP	Copula
NN	Neural Network
INF	Infinitive
AI	Artificial Intelligence
ADJ	Adjective
PRES	Present Tense
PUD	Parallel Universal Dependencies
DEPREL	Dependency Relation
DEPS	Dependencies
IMP	Imperative
PLU	Plural
FORM	Word Form
PART	particle
CARD	Cardinal
NONE	None Of Them
NOUN	Noun
ZERO	Zero
VERB	Verb
NN	Common Noun
A	Agent/Argument
P	Patient/Possessor
PAST	Past

PROG	Progressive Tense
AOR	Aorist
WHILE	While
ABLE	Ability
ROOT	Root

1. INTRODUCTION

Natural Language Processing is a popular development with contributions from computer science, linguistics, and machine learning studies. Natural Language Processing (NLP) is a subfield of artificial intelligence (Artificial Intelligence (AI)) that deals with the interaction between computers and human language. The goal of NLP is to enable computers to understand, interpret, and generate natural language in a way similar to human communication.

Treebanks are one of the cornerstones for any languages in NLP analysis and substantiates the overall performance level even at sentences due to the inaccessibility of a completely stable standardized annotation structure (which also deteriorates accuracy on document /section level named entity recognition, word-sense disambiguation) especially in Turkish. Due to differences in annotation strategies, such as selecting the 'longest root' to make a word the head or opting for the shortest possible root, the model may lose performance in subsequent stages of sentence analysis. For all systems processing CONLLU datasets, this study proposes a method to produce a more optimal annotation structure. By modifying sentence annotations at the word level, the proposed method aims to establish a more comparable annotation format for Turkish dataset analyses, resulting in annotated sentence format. The performance measure of reducing differences in morphological analysis and features will be reference for other future studies, enabling easier work with all datasets.

The Universal Dependencies project aims to develop consistent treebank annotations across languages among many human languages. The annotation scheme is based on Stanford dependencies, Google universal part-of-speech tags, and Interset interlingua for morphosyntactic tag sets. The project is an open community effort with over 600 contributors producing over 200 treebanks in over 150 languages and aims to give a common list of categories and standards for uniformly annotating comparable constructs in

different languages, enabling language-specific modifications as needed.

Universal Dependencies (UD) can be understood as a framework for cross-linguistically consistent morphosyntactic annotation, aiming to support multilingual natural language processing and linguistic studies. The success of UD relies on a delicate balance between multiple factors and it is essential to compromise on these factors, including linguistic analysis for individual languages, linguistic typological consistency, easy and consistent annotation by human annotators, ease of comprehension and use for nonlinguists, high precision computer parsing and assistance with downstream language understanding tasks such as machine translation, relation extraction, and reading comprehension. UD employs a lexicalist approach focused on grammatical relations, which are central to explaining predicate-argument structures across languages. The framework has gained widespread adoption, with treebanks developed for over 100 languages.The project provides annotation guidelines, existing treebanks, and tools for development and exploitation, facilitating its application in NLP and linguistics. While UD strives to optimize these various goals, it remains open to improvement.A habitable design, favoring traditional grammar notions and terminology, is crucial for UD's success. While it is easy to improve UD on one dimension, the challenge lies in maintaining sensitivity to all dimensions. Thus, a successful UD requires a balance between these dimensions.

Stanford dependencies [1] is a backend to the Stanford parser, have been adapted into other languages, and have established to be considered the accepted standard for dependency analysis of English. Using CoNLL-X shared task data and the Google universal tag set, which was developed to translate many tag sets to a single standard, cross-linguistic error analysis has been widely used. [1] The Interset is a tool for converting between morphosyntactic tagsets of many languages that was used in research with cross-lingual delexicalized parser adaptation in 2006. Later, Harmonized Multi-Language Dependency Treebanks (HamleDT) used it as the morphological layer, bringing treebanks

from several languages under a single annotation method.

The first effort to integrate Google universal tags and Stanford dependencies [1] into a universal annotation system was the Universal Dependency Treebank (Universal Dependency Treebank (UDT)) project. The second iteration of HamleDT enabled 30 languages to Stanford /Google annotation in 2014. All of these initiatives were combined into a single cohesive framework to create the new Universal Dependencies, which are based on revised CoNLL-X format (called CONLLU, modified Google universal tag set, updated Interset feature inventory subset, and universal Stanford dependencies. The initial iteration of the new standards introduced an extended universal set of part-of-speech tags, clarifying the definition of categories and allowing for context-sensitive rules or hand correction when converting to UD Part-of-Speech (POS) tags. The Stanford Dependencies model, which is based on grammatical relations-focused description concepts common in various linguistic frameworks, is the ancestor of the dependency representation of UD. Compared to the original Stanford Dependencies (SD), the new taxonomy has fewer relations.

An expanded collection of universal part-of-speech tags was incorporated in the initial edition of the revised rules, which was published in October 2014. This collection serves to clarify the concept of categories by including characteristics that were considered significant by many but were absent from the original proposal. Thus, instead of being merely equivalency classes of categories in particular language treebanks, universal POS categories now have important definitions. This means that context-sensitive rules or human adjustment are frequently needed when converting to UD POS tags. The goal of the UD morphological features is to offer a fundamental collection of characteristics that are important for analysis and are shared by speakers of different languages. Since many linguistic frameworks are predicated on the notion of grammatical relations-focused description, Stanford Dependencies is the source of the dependency representation used

in UD. Concepts like subject, object, clausal complement, noun determiner, and noun modifier serve as the framework for its organization. The new global version aimed to eliminate some of the oddities and English-specific elements from the previous edition while improving relations to better suit the grammatical structures of many languages. Consequently, compared to the original SD, the new taxonomy has fewer relations. [1]

In this study, the introduction section highlights the importance of universal and consistent annotation systems in Natural Language Processing (NLP). It emphasizes the role of the Universal Dependencies (UD) project in creating cross-linguistic standards for morphosyntactic annotations, balancing linguistic diversity and computational precision. Issues in annotation strategies, particularly for languages such as Turkish, are discussed and methods to enhance consistency are proposed. UD builds on frameworks such as Google Universal Tag Set and Stanford Dependencies, aiming to facilitate multilingual NLP and linguistic studies through standardized and adaptable annotations.

In literature review and previous work section, the Turkish treebanks accepted by UD, their features, annotation details, and differences are discussed.

The detailed problem description section discusses the challenges of using Turkish treebanks together due to differing annotation strategies and the lack of morphological analyses in their CONLLU format. It highlights efforts to align these treebanks by comparing annotation differences, converting them into a unified ”annotated sentence” format, and improving morphological and dependency feature labeling for future Turkish NLP tasks.

In methodology and contribution section we discuss the work on identifying annotation differences between Turkish treebanks. The main issue with the data in CONLLU format is the presence of split words without following a consistent rule, where a single sentence may contain multiple split words. The focus is on detecting and addressing these differences across the treebanks.

In the result section, we discussed the identification of differences by comparing the morphological analyses and features generated by our system with those in the Turkish treebanks. We categorized possible errors, missing annotations, and incorrectly marked features, which helped clarify annotation discrepancies. The resulting error categories lay the groundwork for future studies aimed at improving annotations.

In the conclusion section, we improved Turkish treebanks to enhance their use in NLP tasks. The root selection suggestions can serve as valuable resources for future applications, and the study encourages a unified annotation structure across treebanks. With updates to their CONLLU format, these treebanks (BOUN [2] [3], GB [4] [5], IMST [6] [7] and PUD [8] [9]) are now better suited for advanced tasks like dependency parsing. Harmonizing annotation differences through collaboration with other teams will facilitate future integration. The Annotated Sentence format simplifies sentence structure understanding, making treebank comparisons easier and more efficient. The results of the dependency parsing show that treebanks marked with a Starlang structure improve Labeled Attachment Score (LAS), Unlabeled Attachment Score (UAS), and Label Score (LS) scores, enhancing head and dependent words.

These updates will also improve the use of treebanks in Turkish NLP applications by accurately determining word meanings and grammatical features.

2. LITERATURE REVIEW & PREVIOUS WORK

There are nine Turkish treebanks accepted by UD so far and these Turkish treebanks follow different annotation guidelines for UD annotation, leading to uncoordinated correction efforts in Turkish, since version 2.4. [10]

These treebanks, ATIS [11], KeNet [12] , FrameNet [13] , Penn [14] and Tourism [15] are created and annotated by Starlang Software [16] so the pattern used in the marking process was largely consistent due to its consistent methodology.

The primary aim of the study is to lay the foundation for a Turkish FrameNet, which reflects the semantic richness and typological properties of the Turkish language. Turkish FrameNet is built by adopting frames from English FrameNet, with necessary adjustments to accommodate the specific characteristics of Turkish. New frames were introduced where needed, and the project focuses on ensuring compatibility with other Turkish natural language processing resources, such as TRopBank and KeNet, by using the same lexical unit IDs across these resources.

The paper also discusses the strategies and methodologies involved in the annotation process, comparing different approaches used in FrameNet projects for various languages. Turkish, being a low-resource language, presents a unique set of challenges in constructing a FrameNet that can be integrated into NLP applications, such as machine translation and semantic role labeling.

A total of 139 frames and 4080 lexical units (predicates) have been created. These frames include 16 that are unique to Turkish, with the rest derived from English FrameNet. The study emphasizes the close correspondence between Turkish FrameNet and English FrameNet, allowing it to be used in both Turkish and bilingual NLP tasks.

2.1 ATIS

The paper ”Building Annotated Parallel Corpora Using the ATIS Dataset: Two UD-style treebanks in English and Turkish” presents the creation of parallel treebanks using the Air Travel Information System (ATIS) dataset. The dataset, originally compiled for evaluating spoken language systems, has been annotated to provide Universal Dependencies style treebanks in both English and Turkish. These treebanks include human transcriptions of flight information inquiries and consist of 5,432 sentences, with the English treebank containing 61,879 tokens and the Turkish treebank 45,875 tokens.

The main objective of the paper is to build parallel annotated corpora in English and Turkish using the ATIS dataset. This resource will be useful for multilingual natural language processing tasks like machine translation and multilingual parsing, particularly because the data represents spoken language, which is different from the usual written corpora.

For English, the treebank underwent a two-step annotation: part-of-speech tagging and dependency annotation. For Turkish, a five-step process was employed: translation, morphological analysis, morphological disambiguation, POS tagging, and dependency annotation. Turkish’s agglutinative nature adds complexity to the annotation process.

Differences between English and Turkish morphological structures posed challenges in translating and annotating data. Both treebanks were annotated following the Universal Dependencies (UD) framework, which facilitates cross-linguistic comparisons and multilingual parser training.

The Turkish treebank had fewer tokens due to its agglutinative morphology but showed more unique surface forms compared to English. The project contributes two unique treebanks focusing on spoken language and highlights the differences between domain-specific corpora and more general-purpose datasets.

These treebanks represent a valuable linguistic resource due to their focus on spoken language and domain-specific content (air travel inquiries). They offer potential for training NLP systems that require understanding spontaneous speech and domain-specific information. [11]

2.2 KeNet

The "Turkish WordNet KeNet" focuses on the creation and development of KeNet, a comprehensive wordnet for Turkish. The main goal of KeNet is to provide a robust lexical database that includes 76,757 synsets and captures both intralingual and interlingual semantic relations, linking to Princeton WordNet (Princeton WordNet (PWN)). This resource is intended to support natural language processing tasks for the Turkish language, offering valuable insights into its lexicon and semantic relationships.

KeNet was developed to address the need for a more extensive and machine-readable Turkish WordNet compared to previous resources like Turkish (TR)-wordnet of BalkaNet. It includes over 76,000 synsets, capturing lexical relations such as hypernymy, antonymy, meronymy, and more. The creation process involved both human and automated efforts, focusing on building a reliable and comprehensive wordnet for Turkish.

The database was built by extracting and annotating lexical entries from the Contemporary Dictionary of Turkish (Contemporary Dictionary of Turkish (CDT)). Synsets, which group words with similar meanings, were created for various parts of speech like nouns, verbs, adjectives, and adverbs. Further refinement of the synsets was done through merge and split processes to ensure semantic accuracy.

KeNet annotates eight types of intralingual semantic relations, including hypernymy (is-a relationships), antonymy, and holonymy (part-whole relationships). The construction of these relations was based on manual verification against English WordNet (PWN), ensuring accurate cross-lingual mappings.

One major challenge was the lack of a systematic Turkish resource to mark synonymy relations, making it labor-intensive to build synsets from scratch. Another challenge was creating hypernymy relations, especially for Turkish words that didn't have exact counterparts in English Princeton WordNet, which required new synsets to be created for better hierarchical mapping. KeNet also includes mappings to Princeton WordNet, which enable cross-linguistic applications like machine translation. The paper highlights both one-to-one and one-to-many mappings between Turkish and English synsets to account for linguistic differences.

KeNet represents a significant advancement in Turkish lexicography, providing an extensive and semantically rich resource for NLP applications. It has already been integrated into other Turkish NLP projects such as Turkish PropBank (TRopBank) and Turkish FrameNet, facilitating broader use in areas like sentiment analysis, machine translation, and semantic role labeling. The paper underscores the manual effort required due to Turkish's unique morphological and syntactic features, emphasizing the importance of human expertise in creating accurate lexical resources. [12]

2.3 FrameNet

The paper focuses on the process of building Turkish FrameNet, a comprehensive lexicographic resource for Turkish, and the challenges encountered during its development. FrameNet provides semantic information about predicate-argument structures, and the main goal of this project is to create a Turkish version that is compatible with resources like PropBank and WordNet.

The study primarily intends to develop a baseline for a Turkish FrameNet while portraying the semantic diversity and typological aspects of Turkish language. Constructing Turkish FrameNet requires the use of frames from the existing English FrameNet but

systematically modified as per the Turkish structure. New frames were created when necessary, the project aims at the integration of other Turkish NLP resources such as TRopBank and KeNet by using the same lexical unit IDs. FrameNet is a machine learning framework that uses various strategies and methodologies to analyze and interpret data, particularly in the context of Turkish, a low-resource language, which presents unique challenges in integrating it into machine translation and semantic role labeling applications.

The study developed 139 frames and 4080 lexical units, including 16 unique Turkish frames and the rest derived from English FrameNet. The study highlights the close correspondence between Turkish FrameNet and English FrameNet, enabling its use in both Turkish and bilingual NLP tasks. [13]

2.4 Penn

The Penn treebank is ”On Building the Largest and Cross-Linguistic Turkish Dependency Corpus” discusses the creation of a comprehensive Turkish dependency treebank, which is derived from the Penn Treebank. This treebank is significant as it provides a cross-linguistic resource, enabling comparisons between Turkish and English syntactic structures. The primary goal of this project was to build the largest Turkish dependency corpus, consisting of 16,400 sentences translated from the original English Penn Treebank. This corpus is fully annotated for morphology, syntax, and semantics, providing a rich resource for dependency parsing in Turkish.

The morphological and syntactic annotations were performed by a team of linguists, ensuring high-quality manual annotation. The sentences were translated from English and processed using a semi-automatic morphological analyzer developed by the team. Dependency relations were manually annotated using a custom-built interface.

This project offers the first cross-linguistic dependency treebank for Turkish, making it valuable for machine translation and cross-linguistic NLP tasks. The corpus allows for direct comparison between the two languages' syntactic structures, improving the understanding of language-specific characteristics and contributing to the development of dependency parsers.

The different branching typology between Turkish and English required manual adjustments during the translation and annotation processes. Inconsistencies in annotations, such as differences in tagging between linguists, were resolved through team discussions to ensure uniformity.

The final outcome is a Turkish dependency treebank with 16,400 manually annotated sentences, the largest of its kind for the Turkish language. The treebank will serve as a base for future developments, including an automatic dependency parser for Turkish and the automatic conversion of constituency trees into dependency trees. This resource is expected to significantly contribute to Turkish NLP and enhance machine translation capabilities between Turkish and English. [14]

2.5 Tourism

The main objective of the Tourism treebank presented in the paper is to develop an automatic dependency parser. The study aims to create a dependency parser specifically designed for the Turkish language, addressing the unique challenges posed by its typological characteristics.

The research utilizes three distinct data sources: semantic annotation, morphological analysis, and shallow parsing, to conduct precise and coherent dependency analyses. The study applies dependency analyses to a specific database for the tourism sector, analyzing 51,185 words from 20,000 sentences. The study aims to introduce a novel Turkish approach to dependency parsing, contributing to the existing literature in this field.

Overall, the research seeks to enhance the understanding and processing of Turkish language structures through automated methods.The paper presents a novel approach that differs significantly from existing dependency parsing methods in several key aspects.The approach in this study is tailored to the unique typological characteristics of the Turkish language, which differs from languages with which many existing parsers have been developed. The methodology utilizes three distinct data sources; semantic annotation, morphological analysis, and shallow parsing, aiming to improve the accuracy and coherence of dependency analyses, a feature not commonly addressed in existing methods. The research utilizes dependency parsing techniques on a specialized database in the tourism sector, utilizing a corpus of 20,000 sentences, thereby enabling the creation of context-aware parsing rules. The study highlights the development of an automatic dependency annotator, which automates the parsing process using established rules from previous studies, streamlining analysis compared to manual or less automated methods.The proposed approach aims to fill the gap in dependency parsing for Turkish by providing a more specialized, accurate, and automated solution tailored to the language's specific needs.

The Tourism treebank study aims to create an automatic dependency analysis by establishing rules suitable for the structural features of Turkish and to analyze a database specific to the field of tourism. The study utilized semantic annotations, morphological analysis, and surface analyses, conducting a dependency analysis on 20,000 sentences consisting of 51,185 words. The consistency rate of the analysis has been maximized. Additionally, this study has performed statistical analyses and inferences on the structural and linguistic features of Turkish. This work contributes a multi-layered and novel approach to the linguistics literature. [15]

2.6 Customized Turkish Datasets

The other four treebanks exist in the UD 2.14, BOUN [3], GB [5], IMST [7] and PUD [9] are have morphological differences because they were annotated by different teams, taking into account different grammatical features. These treebanks are used during in this study, so their coverage and details explained in the following subsections.

2.6.1 Turkish Grammar-Book Treebank

This Turkish grammar-book treebank, which is comprised of 2,803 sentences or sentence fragments handpicked from Göksel and Kerslake's comprehensive Turkish grammar reference book, has been meticulously annotated in accordance with the Universal Dependencies framework. Out of these entries, 410 are sentence fragments such as example noun phrases. In contrast to the wide range of lengths found in standard treebanks, this corpus has relatively short sentences; it contains a total of 16,516 surface tokens with an average of 5.89 tokens per sentence.The number of inflectional groups (nflectional groups (IG)s), or syntactic tokens used in this corpus is 18,146 and there is a ratio of 1.10 syntactic tokens per surface token. All the sentences have numbered examples from the grammar book and other examples are included where necessary during annotation process. The annotators paid particular attention to sentences that had optional words or phrases.These were listed more than once to show all possibilities for variation. Similarly, each possible analysis was annotated for the sentences that could be read differently depending on interpretation. For morphological analysis of words TRmorph tool was used. Then a basic morphological analyzer resolved the ambiguity of the output of this tool. However, all morphological analyzes were manually verified and corrected for precision.

The sentences tokenized after the morphological analysis were further annotated

using the Brat Rapid Annotation Tool. (BRAT) tool strictly according to UD specifications at that time. In this detailed annotation phase, several features or constructions unique to Turkish were observed to exist beyond what had been covered by extant UD guidelines. These gaps were carefully noted down to facilitate future development of UD guidelines. For instance, the annotation paid special attention to some specific morphological features such as case, voice and modality which are difficult issues in Turkish.

This manual annotation process was important not only for ensuring high accuracy levels but also for understanding the intricate linguistic details involved with Turkish language use. This rigorous method is responsible for a dependable resource for both linguistic research and computational parsing that ultimately leads to better comprehension of Turkish language structures. [4]

2.6.2 Enhancements to the BOUN Treebank Reflecting the Agglutinative Nature of Turkish: BOUN Treebank

The article ”Enhancements to the BOUN Treebank Reflecting the Agglutinative Nature of Turkish” focuses on improving the BOUN Treebank to better capture the linguistic intricacies of the Turkish language, particularly its agglutinative nature. The enhancements mainly involve new annotation conventions to address issues like null morphemes, derivational processes, and syncretic morphemes while adhering to the Universal Dependencies (UD) framework.

The annotation process and details are explained as, firstly, the annotation process began with the manual annotation of raw text using Kanerva et al.'s pipeline tool to create CONLLU files. The initial steps included the automatic annotation of Universal Part-of-Speech (UPOS) tags and certain morphological information. The dependency relations were manually annotated by two native Turkish-speaking linguists to ensure accuracy. Inter-annotator agreement was achieved through the double annotation of randomly selected sentences, resulting in high scores in dependency label match, unlabelled

attachment, and labelled attachment.

To further improve the annotations, a re-annotation process was conducted. This process identified and addressed shortcomings in previous annotations, focusing on derivation and null copula. New strategies were employed to represent derivational morphemes, such as using the Miscellaneous (MISC) tab for the ‘df‘ (derived from) function and splitting lemmas containing the -ki morpheme. Additionally, new functions for null copula annotations were introduced to better reflect their linguistic role in Turkish. Distinctions were made between different functions of the copula ”ol-” and new Extended Part of Speech (XPOS) tags and dependency relations were introduced for more accurate representations.

The annotation process also tackled several challenges. For derivation, derivational morphemes like -lI and -sIz were represented accurately without diverging from the UD framework. To handle null morphemes, new functions for the null copula were introduced in the MISC tab to offer more precise annotations. Furthermore, distinctions were made between the six functions of the copula ”ol-” to improve annotation accuracy.

New annotation conventions were introduced, including new XPOS tags and dependency relations, to better capture the linguistic characteristics of Turkish. These conventions addressed the handling of derivational morphemes and null morphemes, among others. The thorough re-annotation process resulted in significant changes across various annotation fields, including UPOS, XPOS, Dependency Relation (DEPREL), MISC, and Features.

By implementing these enhancements, the BOUN Treebank now offers a more accurate and comprehensive representation of Turkish, facilitating improved performance in natural language processing systems. The re-annotation process involved meticulous manual efforts by linguists to ensure the quality and reliability of the treebank, making it a valuable resource for linguistic and computational research. [2]

2.6.3 IMST: A Revisited Turkish Dependency Treebank

The IMST treebank underwent significant revisions to better capture the unique features of the Turkish language, focusing on various syntactic and morphological details for more accurate and consistent data representation.

The Istanbul Technical University (ITU)-Middle East Technical University (METU)-Sabancı Treebank (IMST) project aimed to enhance the previously existing METU-Sabancı Treebank by re-annotating it to reflect the agglutinative nature of Turkish. This effort involved updating the annotation guidelines and introducing new dependency relations.

The annotation process, which is detailed in this section, was conducted using an updated version of the ITU Annotation Tool. Five annotators, including one linguist and four computer scientists with NLP experience, carried out the task. The team, well-versed in Turkish morphology and syntax, underwent two weeks of supervised training in the new framework before starting the annotation work.

The annotation was improved in a number of significant ways. In order to resolve uncertainties and inconsistencies in the prior annotation approach, new dependency labels were first established. For phrasal arguments, for instance, new terms like argument made it easier to discern between various syntactic functions. An further noteworthy enhancement was the removal of optional annotation. There used to be discrepancies when some characters, particularly punctuation, passed without a head. According to the new approach, every punctuation must be connected, either to the sentence's root node if it begins a sentence, or to the last non-punctuation token before it. The management of omissible tokens, a problem in the prior plan, was also addressed by the revised framework. Web jargon and non-canonical language frequently use them. Dependency labels were added to the new annotation scheme in order to handle these situations without sacrificing consistency.

Standardizing modifiers was yet another important improvement. A single modifier label was used to standardize a number of adjuncts that had previously been assigned conflicting labels. This modification enhanced consistency and made the annotating process simpler.

Reannotation produced substantial modifications, with over 117,732 alterations in various areas. The majority of the modifications ensured more precise morphological data by including the UPOS and XPOS tags.

A more reliable and consistent resource for Turkish dependency parsing is the re-annotated IMST. Subsequent investigations will concentrate on improving the annotation criteria even more and broadening the treebank to incorporate more varied language applications.

The important features of the annotation information in the IMST treebank are highlighted in this overview, with a focus on methodological advancements and their effects on the general consistency and quality of the data.[6]

2.6.4 Turkish Treebanking: Unifying and Constructing Efforts: PUD Treebank

The Turkish PUD Treebank's reannotation and the Turkish National Corpus Universal Dependency (Turkish National Corpus Universal Dependency (TNC-UD)) Treebank's original annotation are described in depth in this work. The goal of these initiatives is to expand and harmonize Turkish universal dependency treebanks in compliance with Turkish grammatical requirements and Universal Dependencies criteria.

A particular emphasis on syntactic relations was placed during the revision of the Turkish PUD Treebank and the first 500 sentences of TNC-UD. To get precise morphological assessments, the researchers combined hand annotations with automated methods, such the Turku Neural Parser Pipeline. Consistency and linguistic depth were ensured by basing the annotation requirements on the UD principles and the Stanford Dependency

system. Two linguists with extensive understanding of Turkish grammar completed the annotation, assisted by four computer scientists and a senior linguist. The publically published criteria, which were developed through cooperative conversations, guaranteed consistency in annotations.

A customized desktop annotation tool with morphological editing and advanced filtering was created and made available as an open-source project. By using a keyboard-driven method, the tool makes annotation more efficient and eliminates the need for mouse interactions. With features like a tabular display, a hierarchical tree structure, and the ease of editing multiword phrases, it is a great option for Turkish.

Linguistic correctness and consistency were the main goals of the Turkish PUD Treebank reannotation. Language-specific syntactic relation tags were simplified, such as compound to nmod:poss and obl:tmod to obl. These were among the notable modifications. These modifications attempted to more accurately represent the grammatical structure and semantics of Turkish phrases.

The impact of re-annotation on parsing precision was assessed through the use of an advanced neural parser that is based on graphs. When paired with extra training data from the TNC-UD Treebank, the re-annotated treebank showed better parsing accuracy, despite a modest fall in attachment scores. This implies that improved parsing performance is a result of consistent and linguistically correct annotations.

In order to increase the TNC-UD Treebank's usefulness for NLP applications, 10,000 sentences must be added to its annotation. Prosodic data, in-depth semantic analysis, and human-validated morphological studies are possible future improvements. [8]

3. DETAILED PROBLEM DESCRIPTION

Turkish treebanks accepted by Universal Dependencies, whose annotation details are mentioned in the previous section, are the basic sources for future Turkish natural language processing operations, but since each treebank uses a different annotation strategy, it becomes difficult to use them together. When these treebanks (BOUN, GB, IMST and PUD) are intended to be used in a more advanced operation such as dependency parsing, a problem arose because these four treebanks only have data in CONLLU format with split words and does not have a morphological analysis. In order to use these treebanks together in the future, they needed to be compared to determine annotation differences and bring them to a common structure. For this purpose, these treebanks needed to be converted to the *annotated sentence* format, the details of it are explained below. The fact that the treebanks currently in the CONLLU format contained split words caused different morphological analysis results, and since each treebank was marked by different teams using different strategies, there were also high differences in the feature labels of the words. Throughout this study, the aim was to ensure that these treebanks could be used as a large dataset in future Turkish natural language processing studies and to lay the groundwork for a common annotation system by improving our automatic morphological analyzer and dependency feature labels.

3.1 Detailed Examples

The problem with the CONLLU format of these datasets is that they contain split words without any specific concrete rule defined. A sentence can sometimes have more than one split word. Below are examples of sentences containing split words. BOUN development dataset ”Faili meçhul cinayet kurbanları için **yıllardır** karanlığa kurşun sıkılıyor.” [3]: yıllar + dır = yıllardır:

Figure 3.1: *Example From BOUN Dataset:yıllar+dır*

ID	FORM	LEMMA	UPOS	XPOS	FEATURES	HEAD	DEPREL
1	Faili	fail	ADJ	Adj	Case=Nom|Number=Sing|Number[psor]=Sing|Person=3|Person[psor]=3	2	amod
2	meçhul	meçhul	NOUN	Noun	Case=Nom|Number=Sing|Person=3	4	nmod:poss
3	cinayet	cinayet	NOUN	Noun	Case=Nom|Number=Sing|Person=3	4	nmod:poss
4	kurbanları	kurban	NOUN	Noun	Case=Nom|Number=Plur|Number[psor]=Sing|Person=3|Person[psor]=3	8	nmod
5	için	için	ADP	PCNom	-	4	case
6-7	**yıllardır**	-	-	-	-	-	-
6	yıllar	yıl	NOUN	Noun	Case=Nom|Number=Plur|Person=3	8	nmod
7	dır	dir	ADP	Since	-	6	case
8	karanlığa	karanlık	NOUN	Noun	Case=Dat|Number=Sing|Person=3	10	obl
9	kurşun	kurşun	ADJ	NAdj	Case=Nom|Number=Sing|Person=3	10	nsubj
10	sıkılıyor	sıkıl	VERB	Verb	Aspect=Prog|Number=Sing|Person=3|Polarity=Pos|Tense=Pres	0	root
11	.	.	PUCT	Punct	-	10	punct

An example containing two split words from IMST [7] test dataset ”Ulusal **Müzedeki** resimler **nasıldı**.”: Müzede+ki = Müzedeki, nasıl+dı = nasıldı.

Figure 3.2: *Example From IMST Dataset: Müzede+ki: Müzedeki, nasıl+dı: nasıldı*

ID	FORM	LEMMA	UPOS	XPOS	FEATURES	HEAD	DEPREL
1	Ulusal	ulusal	ADJ	Adj	-	4	amod
2-3	Müzedeki	-	-	-	-	-	-
2	Müzede	müze	NOUN	Noun	Case=Loc|Number=Sing|Person=3	1	flat
3	ki	ki	ADP	Rel	-	1	case
4	resimler	resim	NOUN	Noun	Case=Nom|Number=Plur|Person=3	5	nsubj
5-6	nasıldı	-	-	-	-	-	-
5	nasıl	nasıl	ADV	Adverb	-	0	root
6	dı	i	AUX	Zero	Aspect=Perf|Mood=Ind|Number=Sing|Person=3|Tense=Past	5	cop
11	.	.	PUCT	Punct	-	5	punct

The another example containing three split words from IMST [7] development dataset ”Özer’lerin Mithatpaşa’daki bahçe içindeki evindeydik.”: Mithatpaşa’da+ki = Mithatpaşa’daki, içinde+ki = içindeki, evinde+ydik = evindeydik.

Figure 3.3: *Example From IMST Dataset:Mithatpaşa’da+ki=Mithatpaşa’daki, içinde+ki=içindeki, evinde+ydik = evindeydik*

ID	FORM	LEMMA	UPOS	XPOS	FEATURES	HEAD	DEPREL
1	Özer'lerin	Özer	PROPN	Prop	Case=Gen|Number=Plur|Person=3	7	nmod:poss
2-3	**Mithatpaşa'daki**	-	-	-	-	-	-
2	Mithatpaşa'da	Mithatpaşa	PROPN	Prop	Case=Loc|Number=Sing|Person=3	7	nmod
3	ki	ki	ADP	Rel	-	2	case
4	bahçe	bahçe	NOUN	Noun	Case=Nom|Number=Sing|Person=3	5	nmod:poss
5-6	**İçindeki**	-	-	-	-	-	-
5	içinde	iç	ADJ	NAdj	Case=Loc|Number=Sing|Number[psor]=Sing|Person=3|Person[psor]=3	7	amod
6	ki	ki	ADP	Rel	-	5	case
7-8	**Evindeydik**	-	-	-	-	-	-
7	evinde	ev	NOUN	Noun	Case=Loc|Number=Sing|Number[psor]=Sing|Person=3|Person[psor]=3	0	root
8	ydik	i	AUX	Zero	Aspect=Perf|Mood=Ind|Number=Plur|Person=1|Tense=Past	7	cop
9	.	.	PUCT	Puct	-	7	punct

To bring all treebank into a common structure, it is necessary to perform an accurate morphological analysis of these four datasets. Another problem encountered is deciding and choosing accurate root of each word but all words in these dataset are annotated by human annotator so it would take immeasurable time by this way. As a solution to this, all sentences in the data set were converted to the annotated sentence format from the CONLLUformat, and each sentence was passed through the morphological disambiguator to determine the correct morphological analysis of each word, but the factor that made this part difficult was that the split words do not have morphological analysis like the whole words and the dependency index references of all words after the split word changed.

3.2 Context and Assumptions

For this work, some restrictions will be introduced because of linguistic assumptions, there are morphological marking differences of sentences in the datasets. There may have been two types of errors after the morphological disambiguation process. First, errors are occurred as a result of the disambiguation process. Second, the CONLLU treebanks may have made morphological errors, but we assume that these are minor. It is observed that in Turkish CONLLU datasets, a root is determined for each word of each sentence, pos tags are marked for each word, and its features are determined. Some of these operations were performed manually by human annotators, while others were performed automatically.The format of the treebank datasets we use in this study is the CONLLU format accepted by Universal Dependencies and explained below the CONLLU Format part.

The *annotated sentence* format is a structure morphologically analyzed and reformatted version of a CONLLUsentence. The table below gives information about the amount of data in Turkish treebanks.

Table 3.1: *Number of Sentences in Turkish Treebanks*

	Dev	Train	Test
BOUN	979	7803	979
GB	0	0	2802
IMST	988	3664	983
PUD	0	0	1000

3.3 CONLLU Format

The revised version of CoNLL-X format, CONLLU, encodes annotations in plain text files using UTF-8, normalized to Normalization Form Canonical Composition, and Line Feed characters (LF) characters. Three types of lines are used; word lines containing annotations, blank lines marking sentence boundaries, and sentence-level comments starting with hash (#), occurring at the beginning of sentences before word lines. The last line of each sentence is a blank line. [17] Sentences are composed of one or more word lines, each containing specific fields.

1. **ID:** Every new sentence begins with the word ID, which is a unique identifier that can be a decimal number for empty nodes or a range for multiword tokens.

2. **Word Form (FORM):** Word form is often referred to as a punctuation sign or surface form.

3. **Base Form of a Word (LEMMA):** A word form's lemma, or stem, is referred to as its root.

4. **UPOS:** Part of speech tag for all those involved. Adposition (ADP) (adposition), Adjective (ADJ) (adjective), Adverb (ADV) (adverb), Auxiliary (AUX) (auxiliary), Coordinating Conjunction (CCONJ) (coordinating conjunctions), Determiner (DET) (determiners), Interjection (INTJ) (interjections), Noun (NOUN) (noun), Number (NUM) (numeral), particle (PART) (particle), Pronoun (PRON) (pronoun), Proper Noun (PROPN) (proper noun), Punctuation (PUNCT) (punctuation),

Subordinating Conjunction (SCONJ) (subordinating conjunction), Symbol (SYM) (symbol), and Verb (VERB) (verb). X:Stands for a other)

5. **XPOS:** Treebank or optional part-of-speech tag which is language-specific. The underscore can be used as a placeholder in its place if it is not available.

6. **FEATURES:** The list of morphological features from the universal feature inventory. If an identified language-specific extension or the list of morphological features from the universal feature inventory is not accessible, it is indicated with an underscore.

7. **Head of a Phrase (HEAD):** A word's head can have an ID value or be zero. (0).

8. **DEPREL:** Acronym for connection of (universal) dependence. The relationship of the UD to the HEAD (root iff HEAD = 0) or a subtype of one that is defined for a certain language.

9. **Dependencies (DEPS):** It offers a thorough list of head-deprel pairings, the Head is an essential part of improving dependency graphs.

10. **MISC:** Any additional annotation.

This is an example sentence from the Turkish BOUN [3] treebank in CONLLU format: "Fakülteyi bitirenler en uçtan göreve başlıyorlarmış."

In UD treebanks, the UPOS, HEAD, and DEPREL columns are not allowed to be unspecified, except in multiword tokens and empty nodes. Underscore (_) is used to denote unspecified values in all fields except ID. The enhanced DEPS annotation is optional in UD treebanks, but if provided, it must be provided for all sentences in the treebank. The processing in rare cases where FORM or LEMMA is the literal underscore is application-dependent.

Figure 3.4: *CONLLU Format Example Sentence*

ID	FORM	LEMMA	UPOS	XPOS	FEATURES	HEAD	DEPREL
1	Fakülteyi	fakülte	NOUN	Noun	Case=Acc|Number=Sing|Person=3	2	obj
2	bitirenler	bitir	VERB	Verb	Case=Nom|Number=Plur|Person=3|Polarity=Pos|Tense=Pres|VerbForm=Part	6	csubj
3	en	en	ADV	Adverb	-	4	advmod
4	uçtan	uç	VERB	Verb	Case=Abl|Number=Sing|Person=3	6	obl
5	göreve	görev	NOUN	Noun	Case=Dat|Number=Sing|Person=3	6	obj
6	başlıyorlarmış	başla	VERB	Verb	Aspect=Prog|Evident=Nfh|Number=Plur|Person=3|Polarity=Pos|Tense=Past	0	root
7	.	.	PUCT	Punct	-	6	punct

3.4 Annotated Sentence Structure

An annotated sentence is an annotated view that might contain several linguistic annotations of a sentence. Sentences of this kind are processed at multiple levels, and the results are used for both linguistic analysis and NLP applications. In the first phase, the sentences in the CONLLU dataset were converted to annotated sentence format and stored for future use. How does the content of annotated sentences usually look, it includes the surface forms of the words in sentence. The morphological components of words (roots, affixes, etc.), including which suffixes attach to the root word and their grammatical functions, such as modifiers, and whether they are transitive, intransitive, or sentential. Here are just a few categories that both humans and machines consider when parsing a sentence for syntactic analysis. Typically, these results are presented as 'trees' in syntax (or more accurately, parse tree) analyses. Semantic information refers to the meaning of words and sentences. An example of an annotated sentence might include the following information: "{turkish=yazıyor }{morphologicalAnalysis=yaz+VERB+POS+ Progressive Tense (PROG)1} {metaMorphemes=yaz+iyor}{namedEntity= None Of Them (NONE)}{propbank=NONE} {shallowParse=VERB} {universalDependency=0$Root (ROOT)}"

Each field and its explanation:

turkish: The surface form of the word.

morphologicalAnalysis: The analysis of the root, grammatical category, and affixes of the word.

metaMorphemes: A more detailed analysis of the morphological components.

namedEntity: Named entity recognition information.

propbank: Semantic role labeling information.

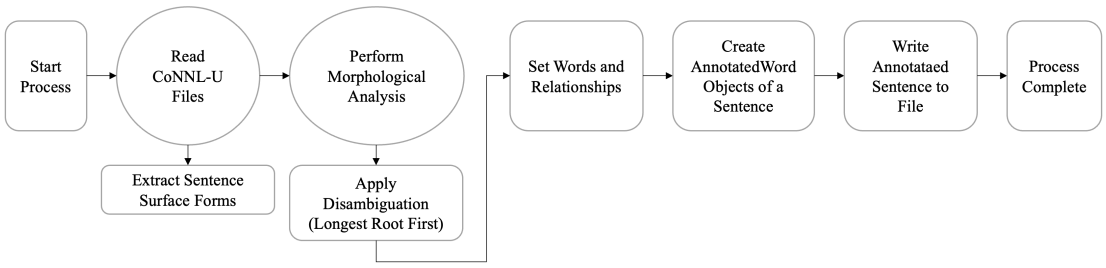
shallowParse: The grammatical role of the word in the sentence.

universalDependency: The position of the word in the dependency tree.

Annotated sentences are used by linguists and NLP researchers to understand the structure and functionality of a language. Such annotations play a critical role in training machine learning models, language analysis, and various NLP applications (e.g., machine translation, text mining, information extraction).

In this process, it reads files according to ”conllu” extension, creates annotated sentence files for each file and saves them in a specific directory. Reading CONLLU File step, the CONLLU file is read using the Universal Dependency TreeBank Corpus class from the Starlang Software [16] ’Annotated Sentence’ package, and the sentences are retrieved.

Figure 3.5: *Conversion From CONLLU Format to Annotated Sentence Format*



Getting Sentence Surface Forms: Surface forms of sentences are obtained by creating Sentence objects.

Morphological Analysis: Morphological analysis of sentences is performed using FsmMorphologicalAnalyzer.

Disambiguation: The correct morphological analysis is determined using the LongestRootFirstDisambiguation class.

Setting Words and Relationships: UniversalDependencyTreeBankWord objects are created and the relationships of each word are set.

Creating AnnotatedWord Object: AnnotatedWord objects are created for each word and these objects are added to AnnotatedSentence.

Writing to File: AnnotatedSentence files are created using the writeAnnotatedSentenceToFile method and saved in the relevant directory.

The pseudocode for converting from CONLLU format to AnnotatedSentence format is given below.

Algorithm 1 Program ConnluToAnnotatedSentenceConverter

```
Set workingDirPath ← GetCurrentWorkingDirectory()
files ← findFiles(workingDirPath, "conllu")
for file in files do
    pureFileName ← RemoveLeadingSlash(GetFileName(file))
    convertFileToAnnotatedSentences(pureFileName, workingDirPath)
end for

Procedure convertFileToAnnotatedSentencesfileName, folderPath
if Not makeDirectory(folderPath + "/" + fileName) then
    return
end if
corpus ← LoadCorpus(fileName)
sentences ← ExtractSentences(corpus)
parsedSentences ← ParseSentences(sentences)
disambiguator ← TrainDisambiguator("penntreebank.txt")
for i = 0 to parsedSentences.size −1 do
    annotatedSentence ← AnnotateSentence(sentences[i], parsedSentences[i], disambiguator)
    fileType ← GetFileType(fileName)
    writeToFile(i, fileType, folderPath, annotatedSentence)
end for
Exit Procedure
Function index, type, path, sentence
fileName ← GenerateFileName(index, type)
Write(path + "/" + fileName, sentence.ToString())
End Function
```

3.5 Universal Features

In Natural Language Processing, a ”feature” is a numerical or categorical representation by which a piece of text (word, sentence, and paragraph) can be analyzed and processed. These features are used to help machine learning models understand text, classify it, and make predictions. Features make raw text more structured and meaningful. Universal Dependecies features indicate additional lexical and grammatical properties of words not covered by POS tags. Universal Features defined by Universal Dependencies are basically in two categories that are attested in more than one language and are considered linguistically important, are Lexical Features and Inflectional Features. [18] Since this thesis focuses on Turkish treebanks, the functions of the features used in Turkish treebanks are explained below.

3.5.1 Lexical Features

The lexical features available in Universal Dependencies [1] indicate specific grammatical properties and enable cross-linguistic comparisons by labeling word-level features. The features specified as lexical features in Universal Dependencies are PronType, NumType, Poss and Reflex. These features are used in morphological and syntactic analyses and help us better understand the grammatical function, category, and meaning of a word.

1. PronType: PronType is the abbreviation stands for ’pronominal type’. This feature generally suitable for pronouns, determiners, pronominal numerals (quantifiers) and pronominal adverbs. [18] The layers of the features and examples corresponding to each relevant layer are provided below.

    - Prs: Personal (e.g. ben “I”, sen “you-Singular (SG)”, o ”he/she/it”, biz ”we”, siz “you-Plural (PL)”, onlar ”they”)

text = **Onlar** bizi farkında değil sanıyorlar. [5]

en = They think we aren't aware of the situation. [5]

1 **Onlar o** PRON _ Case=Nom—Number=Plur—**PronType=Prs** 5 nsubj _ _ [5]

- Rcp: Reciprocal pronoun (e.g birbir- "each other-" and its inflected forms)

  # text = **Birbirinizle** çok iyi anlaşıyorsunuz. [5]

  # en = You get along very well with each other. [5]

  1 **Birbirinizle birbir** PRON _ Case=Ins—Number=Sing—Person=2—**PronType=Rcp** 4 obl _ _ [5]

- Int: Pronoun, determiner, number, or adverb used in interrogation (e.g., kim "who," ne "what," kaç "how many," kimin "whose")

  # text = **Kim** bu insanlar? [9]

  # text_en = Who are they? [9]

  1 **Kim kim** PRON Wh-Pronoun (WP) Number=Sing—**PronType=Int** 0 root _ _ [9]

- Dem: It stands for demonstrative pronoun (e.g o "that", bu "this", şu "this/that")

  # text = **Buraya** araba çarptığını tahmin ediyorum. [5]

  # en = I guess a car bumped here. [5]

  1 **Buraya bura** PRON _ Case=Dat—Number=Sing—**PronType=Dem** 3 obl _ _ [5]

- Ind: It stands for indefinite pronoun (e.g bazı "some")

  # text = Komşularımızın **bazısı** televizyonlarını çok açıyorlar. [5]

  # en = Some of our neighbours turn their televisions on very loud. [5]

  2 **bazısı bazısı** PRON _ Case=Nom—Number=Sing—**PronType=Ind** 5 nsubj

_ _ [5]

- Neg: It stands for negative pronoun or determiner (e.g hiçbiri “none of them” , değil ”not”, hiç “none”, hiçbir “none”)

  # text = Elma **değil**, armut yiyorum. [5]

  # en = I’m eating pears, not apples. [5]

  2 **değil değil** AUX _ **PronType=Neg** 1 aux _ SpaceAfter=No [5]

- Art: It stands for article (e.g bir “a”).

  # text = Çok mu güzel **bir** kız, bu sözünü ettiğin? [5]

  # en = Is she a really pretty girl, this one you’re talking about [5]

  4 **bir bir** DET _ Definite=Ind—PronType=Art 5 det _ _ [5]

2. NumType: NumType is an abbreviation that stands for ’numeral type.’ This feature is generally suitable for numbers that are cardinal, ordinal, or distributive. The question form ’kaç’ (’how many’) is also marked as a number. The layers of this feature and examples corresponding to each relevant layer are provided below.

   - Card: Matching interrogative or cardinal number (e.g 10, 20, 30, on, yirmi, otuz, “ten, twenty, thirty”, kaç elma yedin “how many apples did you eat”)

     # text = **Üç** günden beri hastayım. [5]

     # en = I’ve been ill for three days. [5]

     1 **Üç üç** NUM _ **NumType=Card** 2 nummod _ _ [5]

   - Ord: Ordinal number or corresponding interrogative (e.g 10., 20., 30. onuncu, yirminci, otuzuncu “tenth, twentieth, thirtieth” kaçıncı elmayı yedin “which (of a series) apple did you eat” (answer should be something like birinci “(the) first”))

     # text = - **İkinci** çekmecede olacaklar. [5]

     # en = They’ll be in the second drawer. [5]

2 **İkinci iki** ADJ _ **NumType=Ord** 3 amod _ _ [5]

- Dist: Distributive numeral (e.g üçer, dörder “three each, four each”, kaçar elma yediniz “How many apples did you eat?”)

  # text = **üçer dörder** metrelik on tane heykel [5]

  # en = ten statues, each three or four metres [high] [5]

  1 **üçer üç** NUM _ **NumType=Dist** 3 nummod _ _ [5]

  2 **dörder dört** NUM _ **NumType=Dist** 1 conj _ _ [5]

3. Reflex: Reflex stands for ’reflexive’ and it refers to the subject of its clause. This feature takes boolean value ’Yes’ if it is exist, there is no ’No’ layer exist so it means if it does not seen in features then the word is not reflexive. It covers the inflected forms of the reflexive pronoun ”kendi”.

   - Yes: If it is reflexive put reflex = yes (e.g kendi “self”, kendimiz “ourselves”, kendileri ”themselves”, kendisi “himself/herself/itself”)

   # text = **Kendinizi** anlatmadınız siz hiç. [5]

   # en = You haven’t talked about yourself at all. [5]

   1 **Kendinizi kendi** PRON _ Case=Acc—Number=Sing—Person=2—PronType=Prs—**Reflex=Yes** 2 obj _ _ [5]

3.5.2 Inflectional Features

The inflectional features available in Universal Dependencies [1] indicate specific grammatical properties such as tense, mood, aspect, number, gender, and case due to inflection. This enables the syntactic and semantic roles of words in sentences to be comprehended by providing the necessary information for part-of-speech tagging, parsing, and machine translation. In some languages, some features are marked more than once on the same word. It can be said that there are several layers of the feature. The exact

meaning of individual layers is language-dependent. The features specified as inflectional features in Universal Dependencies and used for Turkish are VerbForm, Mood, Tense, Number, Aspect, Case, Voice, Evident, Polarity, and Person.

1. VerbForm [18]: VerbForm stands for form of verb or deverbative and marked in layers Conv, Fin, Part and Vnoun.

    - Conv: Converbs are non-finite verb forms that combine the characteristics of adverbs and verbs. They are adverbial participles. (For example, Yemek yiyerek kendini daha iyi hissediyor ”She/he feels better by eating,” and izleyerek daha iyi öğreniyor “She/he learns better by watching”)
    - Fin: Verbs indicated for person (Person), tense (Tense), or mood (Mood) are examples of finite verbs that can be mentioned without a conversational context. These verbs have the VerbForm value Fin applied to them. (e.g Okula geldi. “She came school.”, Kütüphaneye geldin sanıyordum. “I thought you came library.”)
    - Part: The verb participle is a non-finite form that functions similarly to an adjective. (e.g izleyeceğim film “the movie that I will watch”, Güler’in ektiği çiçek “the flower that Güler has planted”)
    - Vnoun: Verbal nouns are derived from verbs. (e.g Gülmeni istiyorum. “I want you to laugh”, İniş yapalım. “Let’s land”, Ulaştığını sanmıştım. “I thought you had reached”, Buradan gitmek istemiyordum. “I didn’t want to leave here”)

2. Mood [18]: In finite verbs, mood characterizes the modality, or the speaker’s point of view. Turkish verbs can have more than one mood, and these moods can be expressed in different ways.

- Ind: Stands for indicative.In the indicative mood, a verb just indicates that something occurs, has occurred, or will occur without expressing the speaker's emotion. It is the default mood.(e.g okula gidiyor 'she is going school', müzeye gitti 'she went museum')
- Gen: Stands for generalized modality. This mood is often indicated by the aorist marker on verbs and the -DIr suffix on noun predicates. (e.g sigara içilmez 'one does not smoke = no smoking', bir, bir daha iki eder 'one plus one is two', birin karesi birdir 'one's square is one')
- Imp: Stands for imperative. In Turkish, imperatives are indicated by the lack of any aspect, tense, or modality indicators. Depending on its form, the imperative can be either single, second, or third person plural. Remember that forms that convey a request rather than an order, such as those that are not in the second person singular, may be marked as optional. (e.g dışarı çık 'go out!', hastaneye gidin '(you-Plural (PLU)) go hospital', sinemaya gitsin '(I am ordering him/her to) go cinema!')
- Opt: Stands for optative. In Turkish, the optative suffix (-(y)A) usually expresses a suggestion when associated with first-person identifiers. While it is rare, using second or third person markers suggests a wish. The imperative form, that is more frequently used than -(y)A forms, can also convey a request or proposal with third person singular agreement. (e.g kafeye gidelim 'let's go cafe', inceleyeyim 'let's me examine', gide 'I wish he/she go', gidesin 'I wish you go')
- Nec: Stands for necessitative. This implies a requirement of some kind (in English, must, should, or have to). (e.g hastaneye gitmeli 'she should go hospital', yurtdışına gitmeliydi 'she should have gone abroad')

- GenNec: Stands for general or hypothetical necessitative. It is possible to combine the necessitative suffix with a broad modality suffix. (e.g Ekonomik riskteki artış gözden kaçmamalıdır. “Increase in economic risk should not go unnoticed.”)
- Pot: Stands for potential. A verb can express this mood with taking the suffix -Abil could mean ”possibility” or ”ability.” (e.g okula gidebilir ‘she can go school’ it means ‘she is capable of going school’, or ‘she just may go school’, rüzgar esebilir ‘it may wind’)
- PotPot: Stands for potential expressed twice. There are several verbs that repeatedly express the mood we identify as Pot, especially in negative variants. This often indicates both possibility and ability simultaneously (Something may potentially happen or it might not.). (e.g Gülemeyebileceğini biliyordum. “I knew he/she may not be able to laugh”)
- GenPot: Stands for general or hypothetical potential. A hypothetical assertion or a statement of generalized validity results from the combination of the prospective suffix with the non-past (aorist) suffix. (e.g Sandalyeyi oraya koyabiliriz. “We can/could put the chair there.” (a hypothetical statement))
- GenPotPot: Stands for generallized modality and potential expressed twice. Potential and ability may both be stated in a generic statement or hypothesis. (e.g üniversiteye gidemeyebilir ‘she may not be able to go university’ (it is possible that she is not capable of going university))
- CndGenPot: It represents a conditional predicate with potential and a generalized modality.Conditionality may also be used to a generalization or hypothesis that displays potential or capability. (e.g okula zamanında gidebilirse sınava yetişecek ‘if he/she can go school, he/she will catch the exam’)

- Cnd: Stands for conditional. This suggests that there is a condition. It is the primary method for creating conditionals ('if...') in Turkish. The suffix -sA is what creates this mood. The suffix that comes after Des and Cnd is ambiguous. (okula gittiyse 'if she went school', okula gidiyorsa 'if she is going school', okula giderse 'if she goes school')
- CndGen: Stands for general (non-past) conditional. When the non-past (aorist) suffix is combined with the conditional suffix, the phrase takes on a more expansive predictive meaning that indicates the future. (For example, Balkonu yıkarsa hepimiz rahatlarız. "If he washes the balcony it will be a relief to all of us.")
- CndPot: Stands for conditional potential. It is possible to combine the conditional suffix with the potential modality. (e.g Tartışmaların çoğu konuşarak çözülebilse "if most arguments could be resolved through talking)

3. Tense [18]: Turkish has a complicated system of tenses, aspects, and modes. Turkish verbs can denote past, present, or future acts. Suffixation can also be used to specify complex tenses for activities that took place before to, during, or following a past event. A future event's preconditions, conditions, and consequences are represented by an auxiliary.

   The verbs with value Pqp indicate acts that took place prior to a past reference. Events occurring inside the past reference are referred to as Tense=Past with either the habitual (Hab) or proper progressive (Prog) aspect.

   - Past: Past tense. The Turkish past tense can be expressed on verbal predicates using the suffixes -Past Tense Suffix (DI) or -mIş, and on nominal predicates using the suffixes -(y)DI and -(y)mIş. The difference between the -DI and

-mIş variants is more related to evidentiality than tense. (Evident). Both morphemes allude to an earlier, (fulfilled) occurrence. Additionally, these prefixes are used with others to denote time in relation to a previous occurrence. (e.g hastaneye gitmeliydi ‘she should have gone hospital’, hastaneye gitseydi, ‘if she went hospital”)

- Fut: Future tense. The suffix -(y)AcAk is used to signify the future tense in Turkish. Future Tense morphemes cannot be directly taken by copular predicates. In nominal sentences, the auxiliary ol- is used to represent the future tense. (e.g psikoloğa gidecek ‘she will go psychologist’)
- Pres: Present tense. The absence of past or future morphemes in Turkish indicates the present tense. (e.g psikoloğa gidiyor ‘she is going psychologist’, psikoloğa gitmeli ‘she should go psychologist’, psikoloğa gider ‘she goes psychologist’)
- Pqp: Pluperfect. This indicates a previous activity that had place prior to a reference time. Combining -DI/-mIş and -(y)DI/-(y)mIş allows for this in Turkish; three potential combinations are listed below. The auxiliary ol- is required for nominal predicates in the future tense (hasta olmuştu, ”she had been sick”). Consequently, verbal predicates with double past indications are designated as Pqp. (e.g psikoloğa gitmişti ‘she had gone psychologist’ (this is not evidential), psikoloğa gitmişmiş ‘she had gone psychologist’ (this is evidential), psikoloğa gittiydi ‘she had gone psychologist’ (colloquial))

4. Number [18]: Number is an inflectional feature that denotes agreement with nouns and other speech components, such verbs and adjectives. The characteristic Number appears in verbs and nouns (NOUN, Proposition (PROP), PRON) in Turkish. Remember that the numerical agreement between a subject word and the predicate

is frequently complicated. Plural nouns agree with predicates that have both singular and plural markings, while singular nouns disagree with predicates that have just the plural marking. When the subject is present, the verb in its singular form is preferred. If not, the subject's (and Person's) number characteristic is identified by the appropriate person/number agreement marker.

When a modifier indicates plurality, the noun is not given an explicit plural marker. Only in the presence of an explicit morphological marker do we indicate Number=Plur.

- Sing: Singular. A single individual or thing. There is no morphological indication for singular in the third person singular form of nouns or verbs. (e.g bir çanta 'one/a bag', üç telefon 'three phones' – be careful about missing plural marker. *iki masa"lar" is ungrammatical.)
- Plur: Plural. Multiple individuals or things. On nouns, the suffix -lAr denotes plurality. Plurality is demonstrated by verbs with a set of person/number suffixes that vary depending on the suffixes that come before them. (e.g armutlar 'pears', yedik '(We) ate', Öğrenciler sınava başlayacak(lar) 'The students will start exams' – the plural marker on the verb is optional. , Yemek yiyecekler '(They) will eat meals' – It is assumed that the subject is plural based on the agreement marker on the verb)

5. Aspect [18]: An aspect is a characteristic that indicates how long an activity will take in terms of time, as well as whether it is ongoing or finished.

   A verb's aspect characteristic can be changed in Turkish via a group of verbal morphemes. The Tense and Mood characteristics are frequently impacted by these morphemes as well. There might occasionally be a non-trivial mapping between the aspect values and the suffixes.

- Imp: Imperfect aspect. The action was taken, is taking, or will take some time to complete, the exact date of completion is unknown. (e.g yere gitmeyeceğim "I won't go anywhere")
- Perf: Perfect aspect. The action has been finished or will be finished. (e.g psikoloğa gitti 'she went to psychologist', psikoloğa gitmiş 'she apparently went to psychologist', psikoloğa gidecek 'she will go to psychologist', psikoloğa gitmişti 'she had gone to psychologist (when I arrived)')
- Prog: Progressive aspect. Action is underway with relation to the present or a reference time.Turkish has the progressive markers -(I)yor and -mAktA. The latter is more appropriate in formal settings than the former. It would be challenging to distinguish between the two if not. In rare circumstances, both prefixes may also indicate a habitual characteristic. (e.g psikoloğa gidiyor 'she is going to psychologist (now)', okula gitmekte 'she is going to school (now)', psikoloğa gidiyordu 'she was going to psychologist (when I saw her)', okula gimekteydi 'she was going to school (when I saw her)')
- Hab: Habitual aspect. Verbal morphology in Turkish might indicate a past or present recurrent action. This trait is often indicated with the suffix -A/Ir. On rare occasions, the established quality may be indicated by the progressive suffix -(I)yor. (e.g çok çikolata yer 'she eats lots of chocolate ', kola içerdi 'she used to coke')
- Rapid: It indicates rapid sudden action (new proposal). A particular verb form with the ending -Iver is used to express swift or speedy actions. (e.g oraya gidiver 'quickly go there!', okula gidiverdi 'she immediately/suddenly went to school!')
- Dur: Durative aspect (new proposal). A situation or course of action that has

persisted and is still in progress. This is what we call durative action. The suffixes -Akal, -Agel, and -Adur show this; the first is best characterized as ”durative stative,” while the others are ”durative progressive.” (e.g bakakaldı ‘she looked (for a while, she was frozen while looking)’, yapagelmiştir ‘she have gone on doing (something)’, yiyedur ‘go on eating’)

- Prosp: Prospective aspect. The activity has already occurred or will shortly.The combination of the feature tense marker AcAk and the past tense marker DI in Turkish denotes an impending event. Although it is not as prevalent, the suffix -Ayaz can also be used to indicate future aspect. This form is seldom used with past tense forms and is only seen in a small number of fixed phrases. (e.g gülecekti ‘she was about to laugh’, güleyazdı ‘she was about to laugh’)

6. Case [18]: Case assists in identifying the noun phrase’s function within the sentence. For instance, the verb’s subject and object are frequently distinguished by the nominative and accusative cases, but in languages with set word orders, these distinctions would only be made by the nouns’ placements inside the sentence.

   Case is an inflectional characteristic of nouns in Turkish. When employed as nouns, numerals can occasionally also be inflected for case. This valency attribute is another feature of postpositions; it indicates that the adposition anticipates the condition of its argument.Nominative is sometimes left out of this list. Traditional Turkish is said to include six cases: Nom, Acc, Gen, Dat, Loc, and Abl.

   - Nom: The basic form of the noun, nominative / direct, is usually employed as the citation form (lemma). (e.g Adam yürüyor. “The man walks.”)
   - Acc: Accusative. The definite direct object of a statement is often identified in Turkish using the accusative case. Without the accusative suffix, indefinite

direct objects maintain their original form (Nom). To indicate Acc in Turkish, add the suffix -(y)I (ı/i/u/ü/yı/yi/yu/yü).(e.g Mesajı gördüm “I saw the message”)

- Dat: Dative. The dative case is typically used to indicate advancement into, toward, or to a place or time. The complements (noun phrases) of certain postpositions and their oblique arguments with certain verbs must likewise be in the dative case.The suffix -(y)A (e/a/ye/ya) in Turkish is used to denote dat. (e.g Istanbul’a gidiyorum. “I am going to İstanbul”, Telefonu Esra’ya ver “Give the phone to Esra”, Rüzgara rağmen koşuyorlar “They are running despite the wind” (postposition rağmen “despite” requires a dative complement), Bu davranışa kırıldık “we are broken/upset about the behavior”)
- Gen: Genitive. The English preposition of is frequently used to translate genitive phrases, which have the prototypical meaning of the noun phrase belonging to its governor in some way. Certain postpositional complements must also be in the genitive case. The genitive morpheme also indicates the subject of the subordinate clauses.In Turkish, the suffix -(n)In is used to indicate Gen. (e.g Büşra’nın sunumu “Büşra’s presentation”, Pastayı senin için aldım “I bought the cake for you”)
- Loc: Locative. The locative case is known for its frequent use of place-specific language when expressing time or space. Certain verbs require locative case for their complements (noun phrases) and oblique arguments. Turkish uses the suffix -Locative Suffix (DA) to denote locative. (e.g İşteyim “I am at work”, Sunum dörtte “The presentation is at four”)
- Ins: Instrumental. The term ”instrumental case” comes from the noun’s use as an instrument to carry out an activity. In Turkish, comitative, or the coordination of two sentences, is indicated by the instrumental suffix -(y)lA. Every

usage and interpretation has an Ins symbol. The instrumental suffix must also appear in the oblique arguments of some verbs and in the complements (noun phrases) of some postpositions. Traditionally, instrumental and comitative are not considered instances in Turkish. We characterize as Ins ongoing use of -(y)lA. (e.g İstanbula'a uçakla gitti "she went İstanbul by plane" (instrumental), İstanbula'a Esra'yla giti "she went İstanbul (together) with Esra" (comitative))

- Abl: Ablative. Direction from a certain location is the prototypical meaning. Certain verbs' oblique arguments and some postpositions' complements (noun phrases) also need to be in the ablative. (e.g İstanbul'dan gelmişler "The came from İstanbul", Kalemi Esra'dan aldım "I took/buy the pen from Esra", Ayşe'den hoşlanıyor "He likes Ayşe")
- Equ: Equative. To employ the equative case, one might say something is "X-like," "similar to X," or "same as X". It specifies the standard of comparison as opposed to the equative Degree, which identifies the attribute being compared. The language used is Turkish. (e.g ben "I"; bence "like me")

7. Voice [18]: Verbs have a feature called voice that helps in the mapping of conventional syntactic roles like subject and object to semantic roles like agent and patient.

    - Pass: Passive voice. (e.g Ev temizlendi "The house was cleaned", Buradan kaçılabilir "One may escape from here" (impersonal, intransitive verb passivized))
    - PassPass: Double passive voice. (e.g Burada giyi-n-il-ebilir "One may put on here / this place is good for put on" (stylistic, For impersonal meaning, an intransitive verb just needs to have one passive suffix added, second one is redundant).)

- PassRfl: Combination of passive and reflexive voices. (e.g Burada yıka-n-ıl-dı "[one] washed himself here")
- PassRcp: Stands for a synthesis of reciprocal and passive voices. (e.g Burada yaz-ış-ıl-dı "[two or more people] written here" (lit: [two or more people] write each other here))
- Rcp: Reciprocal voice. (e.g konuştular "they talked (they talked each other)")
- Cau: Causative voice. In causal constructions, the subject is the entity that is "causing" the action. It often translates to "cause/make/have/let/allow" in English, i.e., it instructs the subject to perform the action indicated by the primary verb. Many (lexicalized) verbs with unconnected origins in other languages, such öl- "die" and öl-dür "kill" (to cause someone to die), are connected in Turkish by the causative suffix.
- CauCau: Double causative voice. (e.g Evi temizlet-t-tir-dik "We had someone to have the house cleaned")
- CauPass: Passive causative voice. (e.g Ev Oya'ya temizlet-t-tir-il-di "Somebody caused someone else to cause Oya to clean the house.
- CauRcp: Causative reciprocal voice. (e.g Anneleri kardeşleri barış-tırdı "Their mother made the siblings make peace each other")
- CauPassRcp: Causative reciprocal passive voice. (e.g Onlar konuş-tur-ulmadılar "They were not allowed to talk (lit: they were made (by someone) not to talk each other)")
- Rfl: Reflexive voice. (e.g kaşın "to itch oneself" giyin "to put on oneself")

8. Evident [18]: Evidentiality is a social linguistics notion referring to some type of inflection expressing a speaker's internal source of information, otherwise defining

it as mood and modality. In Turkish, there is a distinction made in the formation of a past tense singular between firsthand and non-firsthand.

- Fh: Stands for evident firsthand. (e.g gitti “he/she/it went” (and I was there and saw them coming))
- Nfh: Stands for evident non-firsthand. (e.g gitmiş “he/she/it has gone” (I did not witness them going but I know it because someone told me / because I see that they are not there now)

9. Polarity [18]: Polarity is a feature in verbs, adjectives, adverbs, and nouns in negating languages. It is used to mark function words with PronType=Neg, unless they are already marked. Positive polarity is rarely encoded using overt morphology. Polarity=Pos is optional for negated words. Pronouns and pronominal parts have no binary opposition like verbs and adjectives.

    - Pos: Positive (e.g ”gitti”: he/she went)
    - Neg: Negative (e.g ”gitmedi”: he/she did not go)

10. Person [18]: Person is a feature in personal and possessive pronouns and verbs, marking the person of the verb’s subject. In some languages, like Basque, it can also mark person of objects. This makes it unnecessary to always add a personal pronoun as subject, leading to pro-drop languages.

    - 1 (first person): The first person singular refers to the speaker or author, while the plural includes the speaker and one or more additional persons. (e.g ”ben, biz”: I, we)
    - 2 (second person): The second person in a sentence refers to the sender of the message, while in a plural context, it may include multiple senders and possibly third parties. (e.g ”sen”: you)

- 3 (third person): The third person refers to individuals who are neither speakers nor addressees. (e.g ”o, onlar”: he, she, it, they)

11. Person[psor] [18]: Person[psor] is possessor’s person. Possessive pronoun + noun forms are used for possessive inflections of nouns, which typically do not have a Person feature. However, possessive morphology typically includes Number, which must be multi-layered on nouns. This layered feature is not used with possessive pronouns, which typically have simple Person. In some languages, possessive pronouns are identical to personal pronouns in the genitive case.

    - 1 (first person possessor) e.g: ”kedim” (my cat), ”kedimiz” (our cat)
    - 2 (second person possessor) e.g: ”kedin” (your cat), ”kediniz” (your cat)
    - 3 (third person possessor) e.g: ”kedisi” (his/her/its cat), ”kedileri” (their cat)

3.5.3 Other Features

Abbr, Typo, Foreign and ExtPos features that cannot be fully included in a category are specified as other features and just Abbr is used in Turkish treebanks which stands for abbreviation. [18]

1. Abbr: The Boolean feature is often abbreviated, but it typically refers to a part of speech other than X. Examples: United States of America (USA), United Kingdom (UK), etc.,

4. METHODOLOGY AND CONTRIBUTION

In this section, we will discuss our work on identifying the annotation differences between Turkish treebanks. The issue with the data in CONLLU format was the presence of words split into two parts without adhering to a specific rule. A sentence sometimes can have more than one splitted word. To identify the differences between treebanks, we performed the following procedures:

1. To facilitate comparisons between datasets, all CONLLU sentences in the four [3] [5] [7] [9] treebanks were converted to the annotated sentence format.

2. By merging split words, new annotated sentences were created for each treebank.

3. By comparing the roots obtained from the morphological analysis of the annotated sentences versus the roots in the CONLLU files from these four treebanks [3] [5] [7] [9], differences were identified in the root annotations of the words.

4. The features of each word in the CONLLU treebanks were compared with their counterparts in the annotated sentence format, and the differences were identified.

5. All affixes found in the second part of split words were extracted to understand the word splitting strategy in these treebanks.

6. Manual morphological corrections were applied to the Annotated Sentence formatted sentences obtained by us, and each treebank was converted back to CONLLU format.

7. For each treebank, the features of every word were generated using our system, and the features written for that word in the CONLLU treebank were obtained. These were then compared to identify words with non-matching features.

8. For each treebank, error types definitions were derived based on words with non-matching features. The feature generation mechanism within the Annotated Sentence generator was then updated to eliminate these error types.

9. This Stanza dependency parser was applied to both two formats, which one is the CONLLU treebanks initially used, and the other is the annotated sentence format, and LAS, UAS, and LS scores were obtained.

The following sections provide a detailed explanation of these processes.

4.1 Annotated Sentence Conversion of BOUN, GB, PUD and IMST Treebanks

The first step in identifying annotation differences between Turkish treebanks was to convert the BOUN [3], GB [5], PUD [9], and IMST [7] treebanks from the CONLLU format to the annotated sentence format. Since the structure of an annotated sentence was explained in the Detailed Problem Description section, this part will directly provide details about the processes through examples.

An example CONLLU sentence from BOUN development dataset [3] is seen the word ”yılındayız” have two part; the first part is ”yılında”, and the second part is ”yız”, two of them combined as a whole word ”yılındayız”, so when it is converted to an annotated sentence, both of the two parts have a morphological analysis as in the annotated sentence format. ”1936 yılındayız.” [3]

Figure 4.1: *Example From BOUN Dataset: ”yılındayız”*

ID	FORM	LEMMA	UPOS	XPOS	FEATURES	HEAD	DEPREL
1	1936	1936	NUM	ANum	NumType=Card	2	nummod
2-3	yılındayız	-	-	-	-	-	-
2	yılında	yıl	NOUN	Noun	Case=Loc|Number=Sing|Number[psor]=Sing|Person=3|Person[psor]=3	4	advmod
3	yız	i	AUX	Zero	Aspect=Perf|Mood=Ind|Number=Plur|Person=1|Tense=Pres	2	cop
4	.	.	PUCT	Punct	-	2	punct

Here is the annotated sentence conversion result for this sentence:
{turkish=1936}{morphologicalAnalysis=1936+NUM+Cardinal
(CARD)}{metaMorphemes=1936}
{namedEntity=NONE}{propbank=NONE}{shallowParse=NUM}
{universalDependency=2$Numerical Modifier (NUMMOD)}

{turkish=yılında}{morphologicalAnalysis=yıl+NOUN+A3SG+P3SG+ Locative (LOC)}
{metaMorphemes=yıl+sH+nDA}{namedEntity=NONE}{propbank=NONE}
{shallowParse=NONE}{universalDependency=0$ROOT}

{turkish=yız}{morphologicalAnalysis=yız+NOUN+A3SG+Possessive Non-Agreement
(PNON)+Nominative (NOM)}
{metaMorphemes=yız}{namedEntity=NONE}{propbank=NONE}{shallowParse=AUX}
universalDependency=2$Copula (COP)}

{turkish=.}{morphologicalAnalysis=.+PUNCT}{metaMorphemes=.}
{namedEntity=NONE}{propbank=NONE}{shallowParse=PUNCT}
{universalDependency=2$PUNCT}

In this example for the CONLLU type, the first part of the word ”yılında” is tagged

as a noun for both UPOS and XPOS, while the second part ”yız” is tagged as an auxiliary for UPOS and zero (i.e., non-existent) for XPOS. The word ”yılında” is marked as the root in terms of dependency relation, and the suffix ”yız” is attached to it with a copula label. In the annotated sentence format, we obtain a more detailed analysis of the wordś structure through its morphological analysis. In this example, the morphological analysis of the word ”yılında” is given as {morphologicalAnalysis=yıl+NOUN+Agent/Argument (A)3SG+Patient/Possessor (P)3SG+LOC}. To explain these tags:

- The root of the word ”yıl” is identified as a NOUN.
- A3SG = agreement 3rd person singular
- P3SG = possessive marker 3rd person singular
- LOC = locative

In Turkish, this translates to the noun ”yıl” with the possessive suffix for the 3rd person singular and the locative case suffix, indicating the word’s form as ”yılında.” The morphological analysis of the ”yız” part of the word is {morphologicalAnalysis=yız+NOUN+A3SG+PNON+NOM}. To explain this analysis:

- ”yız” is taken as the root of the word and identified as a NOUN.
- A3SG = agreement 3rd person singular
- PNON = This tag is used when there are no possessive markers found on the noun. This tag is very important to distinguish certain ambiguities.
- NOM = nominative (This tag indicates that the noun is in the nominative case, which is typically used for the subject of a sentence.)

In summary, ”yız” is analyzed as a noun with 3rd person singular agreement, no possessive marker, and in the nominative case. Since the dependency relation is obtained from the CONLLU dataset, it is marked the same way in the annotated sentence format.

4.2 Annotated Sentence Conversion with Concatenated Words of BOUN, GB, PUD and IMST Treebanks

In the process of finding annotated sentences, the decision of the accurate root is complicated by the split word. The above example CONLLU sentence from BOUN treebank development dataset shows the word ”yılındayız” have two parts, first part is ”yılında”, and the second part is ”yız, two of them combined as a whole word ”yılındayız”, so when it is converted to an annotated sentence, a new stratgy is needed. All split words concatenated as a whole word and the tricky part that was necessary to adjust the correct mappings of dependencies while combining words. This problem was solved by creating a hashmap-based structure that would allow us to set new dependencies by changing the indexes of all dependencies according to the associated word indexes.

The working principle of the algorithm for concatenating a split word is as follows; during the sentence reading process in the CONLLU file, each line is read sequentially, and an Annotated Word object is created and added to the Annotated Sentence object. The two-indexed IDs are assigned sequentially. In this example, for this word

”2-3 yılındayız _ _ _ _ _ _ _ SpaceAfter=No”

the ID is formed by combining the first two IDs, ”2-3”, is taken, with ”2” being used as the ID for the concatenated word and placed in the ID part of Hashmap 1. No Annotated Word object generated from this line, and this line is skipped. The word ”yılındayız” is added to the Hashmap 1 as the value of the ID, to be used as our concatenated word. At this point, the ”decrease Id” value used to reduce word indices starts at 0. When the next line,

”2 yılında yıl NOUN Noun Case=Loc|Number=Sing|Number[psor]=Sing
|Person=3|Person[psor]=3 0 root _ _”
is read, it is checked if the word ID exists in the Hashmap 1. Since the ID = 2 is in the Hashmap 1, the value associated with this key, temporary word surface form ”yılındayız,” is retrieved from the Hashmap 1 and replaced with the word surface form ”yılında.” An Annotated Word object is then created and added.

Each line undergoes these checks to update both the word IDs and the dependency IDs while creating the Annotated Word object, using the keys and values in Hashmap 2, with the IDs decreasing based on the decrease in ID value. For this example sentence, the line starting with ”2-3” is the first line where the split word occurs, and when the word ID = 2 line is read with the initial decrease ID value of 0, no decreas on is made, and the next line is read. At this point, the decrease ID value is incremented by 1. Thus, after each line where a split word is read, the decrease ID is increased, and the indices are shifted correctly by reducing them by the decrease ID amount. Therefore, in the line with word ID = ”3,” the word-ID is decreased by 1 and updated value is ”2” added to the Hashmap 2. The next word ID = 4 is decreased in the same way and added as 3 in the Hashmap 2. When creating the Annotated Words, the indices of word are updated based on this Hashmap 2 after the merging of the split word, ensuring they are correctly adjusted. These updates made to the ID values after the word concatenation process ensure that the words are in the correct line order and the indexes of the dependencies taken from the CONLLU dataset are updated correctly.

Table 4.1: *Split Word IDs and Surface Forms*

Hashmap 1	Word ID	Temporary Word Surface Form
Decrease ID = 0	2	yılındayız

Table 4.2: *Split Word's Second ID and Decreased Form of the ID*

Hashmap 2	IDs Following The Split Word	Decreased Form Of The ID
Decrease ID = 0	2	2
Decrease ID = 1	3	2
Decrease ID = 1	4	3

The resulting Annotated Sentence after word combination process:

{turkish=1936}{morphologicalAnalysis=1936+NUM+CARD}{metaMorphemes=1936}
{namedEntity=NONE}{propbank=NONE}{shallowParse=NUM}
{universalDependency=2$NUMMOD}

{turkish=yılındayız}{morphologicalAnalysis=yıl+NOUN+A3SG+P3SG+LOCˆDerivational Boundary (DB)+ VERB+Zero (ZERO)+Present Tense (PRES)+A1PL}
{metaMorphemes=yıl+sH+nDA+yHz}{namedEntity=NONE}
{propbank=NONE}{shallowParse=NOUN} {universalDependency=0$ROOT}

{turkish=.}{morphologicalAnalysis=.+PUNCT}{metaMorphemes=.}{namedEntity=NONE}
{propbank=NONE}{shallowParse=PUNCT} {universalDependency=2$PUNCT}

Split words can appear more than once in a sentence, so the example below illustrates this condition more clearly by showing how words are concatenated and how word IDs and dependency relation IDs are updated using a hashmap-based shifting index strategy. This example sentence is taken from the IMST [7] development dataset.

According to the algorithm explained in the previous example, hashmaps were obtained for this sentence and the concatenated words were added to the Annotated Sentence in Annotated Word format with the updates of corresponding temporary surface form and corresponding dependency and word IDs.

Figure 4.2: *Example From IMST Dataset: Mithatpaşa'da+ki=Mithatpaşa'daki, içinde+ki=içindeki, evinde+ydik = evindeydik*

ID	FORM	LEMMA	UPOS	XPOS	FEATURES	HEAD	DEPREL
1	Özer'lerin	Özer	PROPN	Prop	Case=Gen|Number=Plur|Person=3	7	nmod:poss
2-3	**Mithatpaşa'daki**	-	-	-	-	-	-
2	Mithatpaşa'da	Mithatpaşa	PROPN	Prop	Case=Loc|Number=Sing|Person=3	7	nmod
3	ki	ki	ADP	Rel	-	2	case
4	bahçe	bahçe	NOUN	Noun	Case=Nom|Number=Sing|Person=3	5	nmod:poss
5-6	**İçindeki**	-	-	-	-	-	-
5	içinde	iç	ADJ	NAdj	Case=Loc|Number=Sing|Number[psor]=Sing|Person=3|Person[psor]=3	7	amod
6	ki	ki	ADP	Rel	-	5	case
7-8	**Evindeydik**	-	-	-	-	-	-
7	evinde	ev	NOUN	Noun	Case=Loc|Number=Sing|Number[psor]=Sing|Person=3|Person[psor]=3	0	root
8	ydik	i	AUX	Zero	Aspect=Perf|Mood=Ind|Number=Plur|Person=1|Tense=Past	7	cop
9	.	.	PUCT	Puct	-	7	punct

Table 4.3: *Split Word IDs and Temporary Surface Forms*

Hashmap 1	Word ID	Temporary Word Surface Form
Decrease ID = 0	2	Mithatpaşa'daki
Decrease ID = 1	5	içindeki
Decrease ID = 2	7	evindeydik

Table 4.4: *Split Word's Second ID and Decreased Form Of The ID*

Hashmap 2	IDs Following The Split Word	Decreased Form Of The ID
Decrease ID = 0	2 : (”Mithatpaşa'da”)	2 : (set ”Mithatpaşa'daki”)
Decrease ID = 1	3 : (”ki”)	2 : (pass the line)
Decrease ID = 1	4 : (”bahçe”)	3 : (update ID of the word ”bahçe”)
Decrease ID = 1	5 : (”içinde”)	4 : (set ”içindeki”)
Decrease ID = 2	6 : (”ki”)	4 : (pass the line)
Decrease ID = 2	7 : (”evinde”)	5 : (set ”evindeydik”)
Decrease ID = 3	8 : (ydik)	5 : (pass the line)
Decrease ID = 3	9 : (”.”)	6 : (update ID of the dot ”.”)

As a result of this algorithm, sentences containing more than one split word are updated with the concatenated forms of the split words for all treebanks and converted

into the annotated sentence format.

4.3 Root Mismatch Detection on CONLLU Datasets and Annotated Sentence

At this stage, differences were detected by comparing the roots marked in the CONLLU dataset with the roots obtained through our morphological disambiguation. Based on these differences, updates were made to ensure the correct root word was selected by the morphological disambiguation process. Since root selection involves a linguistic decision for each word, future studies can focus on enhancements to improve the accuracy of root word selection, aligning with grammar rules.

The root selection process during morphological analysis, that is used in this study, is generally made according to the longest root first strategy. The analyses start with certain possible roots and stems. The main idea is to choose the stem that best suits the meaning of the sentence and the longest one, not to choose the shortest root. The following sentence is an example of this.
”Ali ödevini bitirdi.”
The beginning state of the verb ”bitirdi”, should be marked as the root ”bitir-” because it is a derived verb and has a meaning on its own, instead of the main root ”bit-”.

This rule also applies to nouns. For instance, for a word like ”uçuşlar”, the beginning state should be the nominal root ”uçuş-” instead of the verbal root ”uç-”. However, in a case such as ”uçan”, the derivation will certainly start at the root ”uç”.

If the largest root exists in our lexicon and it fits the meaning of the sentence, it should be selected as the starting state. The words which exist in our lexicon will appear at the beginning of the analysis and their definitions should be selected when the analysis is done. Here are some examples for root/stem selection:

- ”kitaplıkları” $\rightarrow$ the root is ”kitap” but the analysis should start with ”kitaplık” as it

is a part of our lexicon with a separate definition and it is the best suits the meaning of the sentence.

- ”yıkanmıştı” → The root is ”yıka-” but the analysis should start with ”yıkan-” as it is a separate word in our lexicon and it is the best suits the meaning of the sentence.
- ”suçlulukla” → The root is ”suç” but the analysis should start with ”suçluluk- ”as it is a separate word in our lexicon and it is the best suits the meaning of the sentence.

In this part, some roots were changed manually for Annotated Sentence of treebanks to obtain more consistent root selection with CONLLU treebanks from our morphological disambiguator. The table displays some examples.

Table 4.5: *Example Words With Updated Roots*

Treebenk - Annotated Sentence Order	Word	Root from Annotated Sentence	Root from CONLLU Treebank	Updated Annotated Sentence Root
tr_boun-ud-dev.conllu-0048.dev	bana	bana	ben	ben
tr_imst-ud-dev.conllu-0063.dev	sana	sana	sen	sen

The selection of words' roots is carried out by choosing the longest word available in the Turkish Wordnet that best fits the meaning within the sentence. Following the comparison of treebanks' roots, it was decided to continue with this strategy for root selection in our system to achieve consistent results during annotation.

Furthermore, despite improvements in our data through root corrections, as exemplified in the previous table comparing with other treebanks, it was observed that these treebanks did not adhere to a consistent root selection strategy.

For example, in the PUD treebank [9], the root for the word 'güçleri' in the first sentence below is selected as 'güç', while in the second sentence below, the root for

'güçlerinin' is selected as 'güçleri' meanwhile in the BOUN treebank[3] the root for the word 'güçleşiyor' is selected as 'güç' and the root for the 'güçlendirirken' is also selected as 'güç' but these two words represent different meanings.

1. # sent_id = w01096013[9]

   # text = Asya ve Pasifik'te, çoğunlukla Çinli (yaklaşık olarak 7.5 milyon) 3 milyon ve 10 milyondan fazla sivil, Japon işgal ***güçleri*** tarafından öldürüldü.[9]

   23 ***güçleri*** ***güç*** NOUN Common Noun (NN) Number=Plur|Number[psor]=Sing|Person[psor]=3 24 nmod:poss _ _ [9]

2. # sent_id = w02013033[9]

   # text = Süveyş Krizi, değişen bir dünyada Büyük Britanya ve Fransa gibi eski sömürge ***güçlerinin*** sınırlarını gösterdi.[9]s

   14 **güçlerinin** ***güçleri*** NOUN NN

   Case=Gen|Number=Plur|Number[psor]=Sing|Person[psor]=3 15 nmod:poss _ _ [9]

3. # sent_id = ins_271[3]

   # text = Ama bazen de bu oyunu oynamak gerçekten ***güçleşiyor***. [3]

   8 ***güçleşiyor güç*** VERB Verb [3]

   Aspect=Prog|Case=Nom|Number=Sing|Person=3|Polarity=Pos|Tense=Pres 0 root _ SpaceAfter=No [3]

4. # sent_id = pop_958 [3]

   # text = Mars enerjisini Merkür'e aktararak iletişimi ***güçlendirirken***, Merkür Mars'ın zihinselliğini alarak entellektüel bir boyut kazanacaktır.[3]

   6 ***güçlendirirken güç*** VERB Verb

   Aspect=Hab|Case=Nom|Mood=Imp|Number=Sing|Person=3|Polarity=Pos [3]

   |Tense=Pres|VerbForm=Conv|Voice=Cau 15 advcl _ SpaceAfter=No [3]

The root is selected for words ’güçlerinin’ and ’güçleri’ as ’güç’, for word ’güçleşiyor’ as ’güçleş’ and for word ’güçlendirirken’ as ’güçlen’ by using morhological disambiguator in our study that the longest word in the lexicon which best fits the meaning of the sentence. (Since the morphological analysis is quite long, sections from the analysis are given so that the selected root samples can be easily seen.)

1. {turkish=Japon}{morphologicalAnalysis=japon+ADJ} {turkish=işgal} {morphologicalAnalysis=işgal+NOUN+A3SG+PNON+NOM} {***turkish=güçleri***} {morphologicalAnalysis=***güç***+ADJ^DB+NOUN+ZERO+A3PL+P3PL+NOM} {turkish=tarafından}{morphologicalAnalysis=tarafından+ADV} (AS:0614.test)

2. {turkish=sömürge} {morphologicalAnalysis=sömürge+NOUN+A3SG+PNON+NOM} {turkish=***güçlerinin***} {morphologicalAnalysis=***güç***+ADJ^DB+ NOUN+ZERO+A3PL+P2SG+Genitive (GEN)} {turkish=sınırlarını} {morphologicalAnalysis=sınır+NOUN+A3PL+P2SG+Accusative (ACC)} {turkish=gösterdi}{morphologicalAnalysis=göster+VERB+POS+Past (PAST)+A3SG} (AS:0907.test)

3. {turkish=gerçekten}{morphologicalAnalysis=gerçekten+ADV} {turkish=***güçleşiyor***}{morphologicalAnalysis=***güçleş***+VERB+POS+PROG1+A3SG} (AS:0703.test)

4. {turkish=iletişimi}{morphologicalAnalysis=iletişim+NOUN+A3SG+PNON+ACC} {turkish=***güçlendirirken***} {morphologicalAnalysis=***güçlendir***+VERB+POS+Aorist (AOR)^DB+ADV+While (WHILE)}

As demonstrated in the examples, lack of a consistent method in selecting the roots of words derived from derivational or inflectional affixes may affect the efficient use of these treebanks in future studies. Therefore, It is recommended preferable used the longest root word selection strategy that is most consistent with sentence meaning, as developed and utilized within the scope of this study.

The following algorithm was used for detecting root mismatches in CONLLU data sets and processing annotated sentences.

Algorithm 2 Program MismatchReportGenerator

```
Set projectPath ← GetCurrentWorkingDirectory()
annotatedFolders ← findAnnotatedFolders(projectPath)
conlluFiles ← findConlluFiles(projectPath)
outputFilePath ← "_mismatch_report.txt"
for annotatedFolderPath in annotatedFolders do
    for conlluFilePath in conlluFiles do
        if annotatedFolderPath contains datasetType and conlluFilePath contains datasetType then
            nestedMap ← parseAndStoreConlluFile(conlluFilePath, fileType)
            annotatedCorpus ← createAnnotatedCorpus(annotatedFolderPath, fileType)
            writer ← OpenFileWriter(datasetType + outputFilePath)
            compareRoots(annotatedFolderPath, annotatedCorpus, nestedMap, writer, fileType)
            CloseFileWriter(writer)
        end if
    end for
end for
Print "Mismatch reports generated for each dataset."
Exit Program
Function parseAndStoreConlluFileconlluFilePath, fileTypePattern returns Nested Map
sentences ← ParseConlluFile(conlluFilePath,fileTypePattern)
return CreateNestedMap(conlluFilePath, sentences)
End Function
Procedure compareRootsannotatedFolderPath, annotatedCorpus, nestedMap, writer, fileTypePattern
for i ← 0 to annotatedCorpus.sentenceCount() −1 do
    annotatedSentence ← GetSentence(annotatedCorpus, i)
    annotatedRoots ← extractRoots(annotatedSentence)
    annotatedWholeWords ← extractWholeWords(annotatedSentence)
    for conlluSentence in nestedMap do
        for word in annotatedRoots do
            if word not equal to GetWordRoot(conlluSentence, word) then
                WriteToFile(writer, "Mismatch: " + word)
            end if
        end for
    end for
end for
End Procedure
```

Algorithm 3 Main Algorithm for Processing Annotated Corpus and Conllu Files (Only given for BOUN dataset for brevity.)

```
Procedure processCorpusAndConlluFiles(annotatedFolders, conlluFiles, outputFilePath)
if annotatedFolders is empty or conlluFiles is empty then
    Print "Annotated folders or CONLLU files not found."
    Return
end if
for annotatedFolderPath in annotatedFolders do
    for conlluFilePath in conlluFiles do
        if annotatedFolderPath contains "tr_boun-ud-dev" and conlluFilePath contains "tr_boun-ud-dev" then
            nestedMap ← parseAndStoreConlluFile(conlluFilePath, ".dev")
            annotatedCorpus ← createAnnotatedCorpus(annotatedFolderPath, ".dev")
            writer ← createBufferedWriter("tr_boun-ud-dev" + outputFilePath)
            compareFeatures("tr_boun-ud-dev", annotatedCorpus, nestedMap, writer, ".dev")
            close(writer)
        end if
    end for
end for
End Procedure
```

4.4 CONLLU Datasets Sentence and Annotated Sentence Feature Comparison

The features indicate additional lexical and grammatical properties or they are numerical, categorical representation of words not covered by POS tags. This was previously explained in the detailed problem description section.

In order to compare the features in CONLLU Datasets sentences and annotated sentences, features were obtained for each word in annotated sentences with the ”getUniversalDependencyFeatures()” method, which was previously implemented in the ’Annotated Sentence’ package in our system. The features corresponding to the same word marked in the CONLLU dataset were also obtained. These two feature groups were compared without case sensitivity and the features obtained from Annotated Sentence that were not marked in CONLLU or vice versa and that were different, i.e. did not match, were determined and these mismatched features were recorded in an output file named after each dataset.

For the implementation detail of this comparison algorithm, a 4-dimensional linked hashmap was used, which stores the CONLLU datasets as a corpus and allows the order of the sentences read from the datasets to be known. A separate linked hashmap was created for each CONLLU dataset and a structure equivalent to the Annotated Corpus was provided. Annotated Corpuses were created for the Annotated Sentences. In this way, mismatch reports were generated for each Annotated Corpus and each 4-D linked hashmap CONLLU Dataset sent as a parameter to the comparison method. Separate feature lists were generated for each word in an Annotated Sentence read from the Annotated Corpus, these lists were sorted alphabetically, combined and compared with the features already present in the CONLLU datasets, and the differences were determined and added to the mismatch report file created for each dataset.

5. RESULT

Within the scope of this study, differences between annotations were identified by comparing the morphological analyzes and features generated by our system with those marked on Turkish treebanks. Possible errors, missing annotations, or incorrectly marked features were classified. This approach made annotation differences more comprehensible and the resulting error categories provided a foundation for future studies aimed at improving annotations.

5.1 Error Types Categorization For Features

Error types were categorized by grouping the mismatched features identified through feature comparisons into specific categories. The situation expressed as an error here may be that certain features are not labeled at all in our system or are labeled as a different feature for the same word in conflicted datasets, and the opposite is also true. At this stage, it was assumed that the errors made in the features marked in other treebanks were at a minor level, and our universal dependency feature finder algorithm was revised and error type elimination was performed.

After these operations, we collected the errors under four headings to cover their general outlines: Root differences, Inflectional group difference, feature differences, and annotating errors.

5.1.1 Root Differences

The root differences category includes words whose roots are marked differently in the Turkish treebanks used in this study and in our morphological analyzer. Choosing a different root word also causes the features of the word to be marked differently.

Example from BOUN [3] dataset the root of ”bekletin” is choosen as ”bekle-” but it is seen that in our mophological analyser the root is choosen as ”beklet-” so this kind

of different root selected words can have different feature. In here different features are Evident=fh|Tense=pres|Verbform=fin of ’beklet-’ and Voice=Cau for ’bekle’.

sent_id = pop_938 [3]

text = 30 dakika **bekletin**. [3]

3 **bekletin bekle** VERB Verb

Mood=Imp|Number=Plur|Person=2|Polarity=Pos|Voice=Cau **0 root** _ [3]

{turkish=bekletin}{morphologicalAnalysis=**beklet**+VERB+POS+Imperative (IMP)+A2PL}

{metaMorphemes=beklet+yHn}{shallowParse=VERB}{universalDependency=0$ROOT}

(Annotated Sentence 0732.train)

In addition, some examples of root differences are shown in the table below.

Figure 5.1: *Root Differences Examples*

DATASET	Starlang Root	Starlang Features	Dataset Root	Dataset Features
BOUN (Senato'daki Cumhuriyetçilerin oto sektörünün kurtarılmasına karşı çıkması, tasarının yasalaşması ihtimalini **azaltsa** da geçen ayki seçimde yeni başkan olarak seçilen Barack Obama'nın, hazırladığı ekonomiyi iyileştirme planının en az 2.5 milyon kişiye iş imkânı yaratacağı yönündeki açıklamaları ile bazı ülkelerdeki önlemler, borsalara moral verdi.)	azalt	**Evident=fh**|Mood=des|Number=sing|Person=3|Polarity=pos|**Tense=pres**|**Verbform=fin**	azal	Mood=Des|Number=Sing|Person=3|Polarity=Pos|**Voice=Cau**
GB (Bütün atık suları denize **akıttılar**)	akıt	Aspect=perf|**Evident=fh**|Mood=ınd|Number=plur|Person=3|**Polarity=pos**|Tense=past|Verbform=fin	ak	Aspect=Perf|Mood=Ind|Number=Plur|Person=3|Tense=Past|VerbForm=Fin|**Voice=Cau**
IMST (Fesada **aldırmadık**.)	aldır	Aspect=perf|Evident=fh|Mood=ınd|Number=plur|Person=1|Polarity=neg|Tense=past|Verbform=fin	al	Aspect=Perf|Mood=Ind|Number=Plur|Person=1|Polarity=Neg|Tense=Past|Voice=Cau
PUD (1912'de ilk film şirketi (Athina Film) ve 1916'da Asty Film **kuruldu**.)	kurul	Aspect=perf|Evident=fh|Mood=ınd|Number=sing|Person=3|Tense=past|Verbform=fin	kur	Aspect=Perf|Mood=Ind|Number=Sing|Person=3|Tense=Past|Voice=Pass

5.1.2 Inflectional Group Difference

Inflectional Group is an important linguistic unit in Turkish natural language processing studies. Since Turkish is an agglutinative language, words can gain new

meanings and take on different grammatical functions by taking different suffixes. Inflectional Group refers to the group of suffixes added to the root or stem of a word, and each IG is a unit formed by suffixes that have a grammatical function.

A word can have several IGs, and each IG is formed by suffixes added to the previous IG. This is of great importance for understanding and analyzing the morphological structure of Turkish. Therefore, while analyzing the word in this section, we observed that the selection of different inflective groups caused differences in the features of the words. This situation can be explained through an example sentence from PUD [9] dataset.

text = Ayrıca, Paris anlaşmasındaki bir hükmü kullanarak pakttan **ayrılmak** için ilk görev süresinin dolmasını bekleyebilirdi. [9]
text_en = He could also wait until the end of his first term to use a provision in the Paris agreement to pull out of the pact. [9]
10 **ayrılmak ayrıl VERB** Verb-Noun (VN)
Aspect=Perf|Mood=Ind|Number=Sing|Person=3|Tense=Fut|VerbForm=Vnoun
16 xcomp _ _ [9]

In this sentence, the word ”ayrılmak” is a derived word and, as we explained, every word group formed by taking a derivational suffix is an inflective group. It is seen that this word is annotated with VerbForm=Vnoun (verbal noun) feature.

The morphological analyze of the word ”ayrılmak” is {turkish=ayrılmak} {morphologicalAnalysis=ayrıl+VERB+POSˆDB+NOUN+Infinitive (INF)+A3SG+PNON+NOM}
{metaMorphemes=ayrıl+mAk}{shallowParse=VERB}{universalDependency=15$Open

Clausal Complement (XCOMP)}. (Annotated Sentence 0094.test)

When this morphological analysis is examined;

- ayrıl: Verb Root
- VERB: Verb
- POS: Positive
- DB: A morphological boundary at the end of a word (word boundary, derivational boundary)
- NOUN: Noun. This shows that the word has become a noun as a result of the derivation process.
- INF: Infinitive suffix. Indicates that it is derived with the infinitive suffix.
- A3SG: Third person singular suffix
- PNON: No possessive suffix
- NOM: Nominative case. Indicates that the word is in its bare form (without any suffixes).

The different feature from morphological analyze is ”case=nom” so our analyzer behaves like this word is ’noun’ after and the different features from PUD dataset are aspect=perf|mood=ınd|tense=fut|verbform=vnoun. Verbal Noun feature also states that this word is a derived word. As can be seen here, different inflective group selection during annotation processes causes feature differences between words. Such situations are a matter of discussion for bringing Turkish annotation standards to a common ground. The tables display examples for inflectional differences.

Figure 5.2: *Inflectional Group Difference Examples-1*

DATASET	Starlang Root	Starlang Features	Dataset Root	Dataset Features
BOUN (Köy düğünlerinde gelinin ata binmesi önemli bir **adetmiş**.)	adet	**Aspect=perf** | Evident=nfh | **Mood=ınd** | Number=sing | Person=3 | Tense=past | **Verbform=fin**	âdet (adetmiş âdet VERB Verb)	**Case=Nom** | Evident=Nfh | Number=Sing | Person=3 | Tense=Past
BOUN (Doğru yönlendirilmedikleri takdirde, ülkemiz bugün içinde bulunduğu durumdan çok daha büyük zorluklarla karşı karşıya kalacaktır.)	yönlendir	**Case=nom | Number=plur | Number[psor]=sing | Person=3 |** Person[psor]=3	yönlendir	**Aspect=Perf | Number[psor]=Plur |** Person[psor]=3 **| Polarity=Neg | Tense=Past | VerbForm=Part | Voice=Pass**
BOUN (Belki yüzyıl yaşayan birinden daha çok soluk alıp vermek zorunda kaldı; dünyanın temiz havası yetersizdi, tüm dünya temiz havaya kavuşunca astım dinebilir, ruh dinginliğine **kavuşabilirdi**.)	kavuş	**Aspect=perf |** Evident=fh **| Mood=genpot |** Number=sing **|** Person=3 **|** Polarity=pos **| Tense=past | Verbform=fin**	kavuş	**Aspect=Hab** | Evident=Fh | **Mood=Pot** | Number=Sing | Person=3 | Polarity=Pos | **Tense=Pres**
GB (Erken emeklilik, kendi **isteğimdi**.)	istek	**Aspect=perf | Evident=fh | Mood=ınd** | Number=sing | Number[psor]=sing | **Person=3** | Person[psor]=1 | **Tense=past | Verbform=fin**	istek	**Case=Nom** | Number=Sing | Number[psor]=Sing | Person[psor]=1

Figure 5.3: *Inflectional Group Difference Examples-2*

DATASET	Starlang Root	Starlang Features	Dataset Root	Dataset Features
GB (paramızın yüzde doksan **beşi**)	beş	**Case=nom** | Number=sing | Number[psor]=sing | **Person=3** | Person[psor]=3	beş	Number=Sing | Number[psor]=Sing | **NumType=Card** | Person[psor]=3
GB (**buralarda**)	bura	Case=loc | Number=plur | **Person=3**	bura	Case=Loc | Number=Plur | **PronType=Dem**
GB (Zehra torununu görmek **istediği** için Bursa'ya uğradık.)	iste	**Case=nom |** Number=sing **| Number[psor]=sing | Person=3 |** Person[psor]=3	iste	**Mood=Ind** | Number=Sing | Person=3 | **Tense=Past | VerbForm=Fin**
GB (Amerika'ya giderse Jale onunla **gitmeyebilirdi**.)	git	**Aspect=perf | Evident=fh** | Mood=genpot | Number=sing | Person=3 | Polarity=neg | Tense=past | Verbform=fin	git	**Aspect=Hab** | Mood=GenPot | Number=Sing | Person=3 | Polarity=Neg | Tense=Past | VerbForm=Fin

Figure 5.4: *Inflectional Group Difference Examples-3*

DATASET	Starlang Root	Starlang Features	Dataset Root	Dataset Features
IMST (Aslında, balıkçıların Saffet'ten memnun **olmadıkları** da söylenemezdi.)	ol	**Case=nom|Number=plur|Number[psor]=sing|Person=3**|Person[psor]=3	ol	**Aspect=Perf|Mood=Ind|Number[psor]=Plur**|Person[psor]=3|**Polarity=Neg|Tense=Past|VerbForm=Part**
IMST (Benim gibi yüksek tansiyonu olanlar tuzsuz da **yer**.)	ye	**Aspect=hab|Evident=fh|Mood=gen**|Number=sing|**Person=3**|Polarity=pos|Tense=pres|**Verbform=fin**	yer	**Aspect=Perf|Mood=Imp**|Number=Sing|**Person=2**|Polarity=Pos|Tense=Pres
PUD (Primli Tahvillere ve diğer Ulusal Tasarruf ve Yatırım hesaplarına yatırılan paralar, Devlet harcamalarını finanse **etmek** için kullanılır.)	et	**Case=nom**|Number=sing|Person=3	et	**Aspect=Perf|Mood=Ind**|Number=Sing|Person=3|**Tense=Fut**

5.1.3 Feature Differences

Feature differences can indicate features that are marked differently across datasets, or features that are marked in one dataset but not in other datasets. During this study, we found that there were a large number of words whose features were marked differently or that were not marked at all for a particular feature, and we made corrections to our annotated sentences by referencing other datasets. By giving an example from IMST[7] dataset, make this situation more understandable.

sent_id = mst-5582[7]

text = Gülce, incecik de olsa altın bilezik alabiliriz, diyor.[7]

8 **alabiliriz al** VERB Verb [7]

Aspect=Hab|**Mood=Pot**|Number=Plur|Person=1|Polarity=Pos|Tense=Pres 10 obj _ SpaceAfter=No [7]

The morphological analyze of the word ”alabiliriz” is {turkish=**alabiliriz**} {morphologicalAnalysis=**al**+VERB+POSDB+VERB+Ability (ABLE)+AOR+A1PL}

{metaMorphemes=al+yAbil+Hr+yHz} {shallowParse=VERB}
{universalDependency=10$Object (OBJ)} (A.S 3629.train)

The different features from morphological analyze are mood=genpot|verbform=fin and the different features from IMST dataset is mood=pot fot the word ”alabiliriz”. The verbs indicated for mood (Mood), tense (Tense), or person (Person) are finite and are assigned the VerbForm value Fin. The verbform=fin stands for finite verb. so it is need to add this feature to IMST dataset features.

On the other hand, mood=gent pot stands for general or hypothetical potential. A hypothetical assertion or a statement of generalized validity results from the combination of the prospective suffix with the non-past (aorist) suffix.

Whereas mood=pot is stands for potential it means that The suffix -Abil could suggest potential or ability. but it can be hard to distinguish these features sometimes because of the mean of the sentence. A decision should be made in terms of grammar and such features should be updated.

Figure 5.5: *Feature Difference Examples-1*

DATASET	Starlang Root	Starlang Features	Dataset Root	Dataset Features
BOUN (Bunu her şeye karşın yapamıyor **muyuz**?)	mi	Number=plur|Person=1|**Prontype=int**|Tense=pres	mi	**Aspect=Imp**|Number=Plur|Person=1|Tense=Pres
IMST (O kendi götüremez **miymiş**.)	mi	Number=sing|Person=3|**Prontype=int**|Tense=past|**Verbform=fin**	mi	**Aspect=Perf|Evident=Nfh|Mood=Ind**|Number=Sing|Person=3|Tense=Past
BOUN (Güvenlik kamera setinizi kurduktan sonra çevrimiçi olarak ücretsiz bir hesap aça**bilirsiniz.)**	aç	Aspect=hab|**Evident=fh|Mood=genpot**|Number=plur|Person=2|Polarity=pos|Tense=pres|**Verbform=fin**	aç	Aspect=Hab|**Mood=Pot**|Number=Plur|Person=2|Polarity=Pos|Tense=Pres
BOUN (Kızım **benden** yemek istedi.)	ben	**Case=abl|Number=sing|Person=1|Prontype=prs**	bende	**Case=Nom|Number=Sing|Number[psor]=Sing|Person=3|Person[psor]=2**
IMST (Ramiz hala **anlatmakta** diretiyor: Evde yemek götürecek kimse yoktu, diyor, sonra annem bana...)	anlat	Aspect=prog|**Evident=fh**|Mood=ınd|Number=sing|Person=3|Polarity=pos|Tense=pres|**Verbform=fin**	anlat	Aspect=Prog|Mood=Ind|Number=Sing|Person=3|Polarity=Pos|**Polite=Form**|Tense=Pres

Figure 5.6: *Feature Difference Examples-2*

DATASET	Starlang Root	Starlang Features	Dataset Root	Dataset Features
IMST (Risk alabilen yatırımcı seçime yaklaşıldıkça pozisyon **alabilir.**)	al	Aspect=hab|**Evident=fh**|**Mood=genpot**|Number=sing|Person=3|Polarity=pos|Tense=pres|**Verbform=fin**	al	Aspect=Hab|**Mood=Pot**|Number=Sing|Person=3|Polarity=Pos|Tense=Pres
IMST (**Benim** de sana Tibet'ten bahsetmemin tam zamanı.)	ben	**Case=gen**|Number=sing|**Person=1**|**Prontype=prs**	ben	**Case=Nom**|Number=Sing|**Number[psor]=Sing**|**Person=3**|**Person[psor]=1**
PUD (Ülkenin batı kısmından **farklılaşır**; çünkü belirgin topografik özellikleri kıyıya paralel değildir.)	farklılaş	Aspect=hab|**Evident=fh**|**Mood=gen**|Number=sing|Person=3|**Polarity=pos**|Tense=pres|**Verbform=fin**	farklılaşır	Aspect=Hab|**Mood=Ind**|Number=Sing|Person=3|Tense=Pres
PUD (**Ben** zaten her şekilde hapse gireceğim, umarım buna değmiştir.)	ben	Case=nom|Number=sing|Person=1|**Prontype=prs**	ben	Case=Nom|Number=Sing|Person=1|**Polarity=Pos**

5.1.4 Annotation Errors

Annotation errors include incorrectly selecting the feature layer despite deciding on the correct feature label. We can consider errors such as marking the plural word as singular for the Number feature, incorrectly selecting the first, second, and third person for the Person feature, and not adding the abbr=yes feature to a word that is an abbreviation as annotation errors for features. By giving an example from dataset make this situation more understandable.

text = Vuruş **yanlışları** yaptık. 2 **yanlışları yanlış** ADJ Adj Case=Nom|Number=Plur|**Number[psor]=Sing**|Person=3|Person[psor]=3 3 obj _ _

The morphological analyze of the word ”yanlışları” is {turkish=yanlışları} {morphologicalAnalysis=yanlış+ADJDB+NOUN+ZERO+A3PL+**P3PL**+NOM} {metaMorphemes=yanlış+lArH}{shallowParse=ADJ}{universalDependency=3$OBJ} (Annotated Sentence 0256.train)

In here our morphological analyzer gives number[psor]=plur. Number[psor]means that it shows if a noun's owner is singular or plural. In this sentence the possessor can be understand from predicate "yaptı-k", ("we did") so in Turkish this suffix indicates plural. But this feature from BOUN dataset number[psor]=sing is given singular so it is not correct choice for this feature. The below examples displays some annotation errors in treebanks.

Figure 5.7: *Annotation Errors Examples-1*

DATASET	Starlang Root	Starlang Features	Dataset Root	Dataset Features
BOUN (Ve işte o, beni kahreden, somurtganlaştıran, saatlerce ağlatan, **gözyaşlarıyla** dolu...)	gözyaş	Case=ıns|Number=plur|**Number[psor]=plur**|Person=3|Person[psor]=3	gözyaş	Case=Ins|Number=Plur|**Number[psor]=Sing**|Person=3|Person[psor]=3
BOUN (O nasıl **derse** desin uğraştığı sanatın kendisine emanet olduğunu söyleyen üstadları vardı.)	de	**Aspect=hab|Evident=fh|Mood=cndgen**|Number=sing|Person=3|**Polarity=pos|Tense=pres|Verbform=fin**	ders	**Case=Dat**|Number=Sing|Person=3
BOUN (Çocukların hepsi birden alaylı kahkahalar atınca, Nasuh Bey **adımlarını** yavaşlattı, durdu.)	adım	Case=acc|Number=plur|Number[psor]=sing|Person=3|**Person[psor]=2**	adım	Case=Acc|Number=Plur|Number[psor]=Sing|Person=3|**Person[psor]=3**
GB (Yeni bir bilgisayar **almam** gerek.)	al	Case=nom|Number=sing|Number[psor]=sing|**Person=3**|Person[psor]=1	al	Case=Nom|Number=Sing|Number[psor]=Sing|Person[psor]=1|**VerbForm=Vnoun**

Figure 5.8: *Annotation Errors Examples-2*

DATASET	Starlang Root	Starlang Features	Dataset Root	Dataset Features
IMST (Bilimsel Devrim'in **kahramanlarıyla** Kilise arasındaki ölüm kalım savaşının artık insanlığın ortak kültürüne mal olmuş örnekleri, bu sürecin ne kadar çetin ve zahmetli bir süreç olduğunu göstermektedir.)	kahraman	Case=ıns|**Number=plur**|Number[psor]=plur|Person=3|Person[psor]=3	kahraman	Case=Ins|Number=Plur|**Number[psor]=Sing**|Person=3|Person[psor]=3
IMST (Bir mağaza çocuk dostu bir yaklaşım sergilemiyorsa müşteri ebeveynler bunu algılar ve bu mağazadan uzak durur.)	algıla	**Aspect=hab|Evident=fh|Mood=gen|Number=sing|**Person=3**|Polarity=pos|Tense=pres|Verbform=fin**	algı	**Case=Nom|Number=Plur**|Person=3
PUD (Demiryolları tarihi üzerine çeşitli kitaplar yazan Christian Wolmar, 1 Aralık'ta yapılacak yarışmaya katılacak.)	1	**Numtype=card**	1	**Number=Sing|NumType=Ord**
PUD (Eski İngiliz Güney Kamerunları 1 Ekim 1961'de Kamerun Federal Cumhuriyeti'ni kurmak üzere Fransız **Kamerun'u** ile birleşmiştir.)	kamerun	**Case=nom**|Number=sing|Number[psor]=sing	Kamerun'	Number=Sing|Number[psor]=Sing|**Person[psor]=3**

The percentage of all errors in the treebanks is shown in the table.

Figure 5.9: *Percentage Errors According to Feature Comparison*

Dataset	Total Mismatch	Total Number of Words	Percentage
BOUN	3810	125 212	3,04%
GB	1124	17 177	6,54%
IMST	6900	58 096	11,87%
PUD	2185	16 881	12,94%

5.1.5 Dependency Parsing Results and Scores

Dependency parsing is a method in natural language processing used to understand the grammatical structure of a sentence. This parsing identifies ”head-dependent”

relationships between words in a sentence, indicating to which word each word is 'dependent' and with what function. Dependency parsing helps in deriving meaning from the structural relationships in language and is widely applied in NLP tasks like semantic analysis, machine translation, and question answering.In dependency parsing, each word is evaluated as either a 'dependent' or a 'head'.

Head word is the top-most element in the sentence in terms of meaning, with other words dependent on it. Dependent word is a word that derives its meaning or function from the head word. In this dependency structure, 'labels' are used to show the relationship between words. For instance; nsubj (nominal subject): denotes the subject word, obj (object); denotes the object word, advmod (adverbial modifier); denotes an adverbial modifier.

In the example sentence 'Cats eat fish'; 'eat' is the head, 'cats' is linked to 'eat' as the subject (nsubj) and 'fish' is linked to 'eat' as the object (obj). In such analyses, each word's head and its type of relationship (deprel) are specified to structure the sentence.

In this study, Stanza Dependency parser was used as the dependency parsing tool. Stanza is a Python-based NLP library developed by Stanford University, supporting a wide range of languages, including Turkish. Stanza's Dependency Parser is trained to analyze dependency structures in sentences by assigning each word a "head" and a "relationship type." Stanza Dependency Parser is provide efficient models and it is trained on comprehensive datasets like Universal Dependencies (UD), offering models adapted to each language's datasets. Stanza's dependency parser is flexible and modular, it means you can run just the dependency parsing module or extend it to grammatical analysis, tagging, and lemmatization. It also supports Graphics Processing Unit (GPU) usage for accelerated processing, which speeds up training and prediction on large datasets.

Dependency Parsing Scores:

There is three main scores, LAS, UAS, and LS Scores are used to interpret dependency

parsing results. These three scores are used to evaluate the performance of dependency parsing, which aims to find the grammatical relationships between words in a sentence and analyze how each word relates to others. These relationships are often visualized by extracting a ”tree structure” for a sentence. The usage of each score is explained below.

LAS: It represents the 'correct connection + correct label.' This score measures if a word is correctly connected to another word (head) and if this connection is labeled with the correct relationship (such as subject or object roles). LAS is used to measure the performance of a model in finding the correct head and assigning the correct label to this connection. It's the most comprehensive score as it evaluates both relationship and labeling accuracy.

UAS: It means 'correct connection only.' This score only measures if a word is connected to the correct ”head” word without considering the label of the relationship. UAS is used to measure the accuracy of just the structural connections, regardless of labels. It only evaluates if words are correctly connected, making it useful when label accuracy is less critical.

LS: It means 'correct label only.' This score only measures if the grammatical relationships of words are correctly labeled (such as subject, object, adverbial), without considering the head. Since this score only evaluates labels, it shows if grammatical relationships are accurately assigned. It reveals how well the model understands the relationships between words.

Score Ranges and Interpretation: These scores are typically expressed between 0 and 1 (or 0-100% if shown as a percentage). A score of 1 or 100% represents perfect performance, meaning all words are correctly connected to other words with accurate grammatical relationship labels. A score of 0 means that no connections or labels are correct.

0.90 and above: Very high performance. The model correctly links and labels most words, often achieved by large and comprehensive language models.

0.75 - 0.89: Good performance. Most connections and labels are accurate, though some errors remain. Further modification of the data may improve performance.

0.50 - 0.74: Moderate performance. There are significant errors in correctly connecting or labeling many words. Additional training data, better model configurations, or other strategies are needed.

Below 0.50: Low performance. The model often misidentifies connections or labels. This indicates a significant issue with the model structure, data, or training process.

If a model has LAS = 0.85, UAS = 0.90, and LS = 0.88, we can infer that the model correctly connects most words to their heads and labels these connections accurately for the most part. If a model has a high UAS score (e.g., 0.90) but a low LAS score (e.g., 0.75), it means that the model is finding the correct structural connections but struggles to assign the correct grammatical labels for these connections. These three scores highlight areas where the model is strong or weak. Especially in complex tasks such as dependency parsing, it is crucial to evaluate a model's ability to both find the correct relationships (UAS) and label these relationships accurately (LS).

Figure 5.10: *The CONLLU Turkish treebanks and Starlang annotated sentence Stanza dependency parsing score*

STANZA DEPENDENCY PARSER	UAS SCORE	LAS SCORE	LS SCORE
BOUN TEST	67.9 -> **68.2**	56.8 -> **56.8**	67.0 ->**67.3**
IMST TEST	67.1 -> **71.9**	57.1 -> **60.3**	68.9 -> **69.7**
GB TEST	70.7 -> **73.9**	57.1 –> **59.1**	65.8 –> **66.1**
PUD TEST	54.4 -> **57.7**	44.2 -> **46.0**	57.5 -> **59.5**

In this study, Stanza dependency parsing applied both pure CONLLU Turkish treebanks and treebanks converted from annotated sentence, which is Starlang sentence structure, to conllu format. The final scores of the parsing process with two different model (BOUN and IMST) can seen tables. The CONLLU Turkish treebanks and Starlang annotated sentence dependency parsing scores are given above.

6. CONCLUSION

In this study, improvements were made to Turkish treebanks, on which no advanced work had previously been done, to enable their more active use in Turkish NLP studies. The suggestions for root selection can serve as fundamental resources for future Turkish natural language processing tasks. Since each treebank employs a different annotation strategy, this study paves the way for the adoption of a unified structure, bringing them together.

It will be more easier and possible to use these treebanks (BOUN [2] [3], GB [4] [5], IMST [6] [7] and PUD [8] [9]) in a more advanced operation such as dependency parsing, with updates on their CONLLU format with concatenated words. In order to use these treebanks together in the future, annotation differences can be brought to a common structure with the contributions of the teams that created other treebanks. The Annotated Sentence format, provides convenience to better understand the structure of each sentence so it may become a gold standard to keep Turkish dataset and the comparison between treebanks much more easier and will be automated so this will save a lot of time. When we examine the dependency parsing results, we see that treebanks marked with Starlang sentence structure can increase LAS, UAS, and LS scores despite the decrease in the number of dependencies by merging the divided words, which means it will contribute to the improvement of head and dependent word marking. Thanks to the fixes to be made for the features they are used in Turkish natural language processing applications to determine the correct meaning of the word, its grammatical function and other grammatical features.

REFERENCES

[1] U. Dependencies, "Universal dependencies." https://universaldependencies.org/, 2024.

[2] B. Marsan, F. Akkurt, M. Sen, M. Gürbüz, O. Güngör, S. B. Özates, S. Üsküdarli, A. Özgür, T. Güngör, and B. Öztürk, "Enhancements to the boun treebank reflecting the agglutinative nature of turkish," *ArXiv*, vol. abs/2207.11782, 2022.

[3] U. Dependencies, "Ud turkish boun." https://universaldependencies.org/treebanks/tr_boun/index.html, 2024.

[4] Çagri Çöltekin, "A grammar-book treebank of turkish," in *A grammar-book treebank of Turkish*, 2017.

[5] U. Dependencies, "Ud turkish gb." https://universaldependencies.org/treebanks/tr_gb/index.html, 2024.

[6] U. Sulubacak, G. Eryiğit, and T. Pamay, "Imst: A revisited turkish dependency treebank," in *Proceedings of TurCLing 2016, the 1st International Conference on Turkic Computational Linguistics* (B. Karaoğlan, T. Kışla, and S. Kumova, eds.), (Turkey), pp. 1–6, EGE UNIVERSITY PRESS, Apr. 2016. 1st International Conference on Turkic Computational Linguistics, TurCLing 2016 ; Conference date: 03-04-2016 Through 09-04-2016.

[7] U. Dependencies, "Ud turkish imst." https://universaldependencies.org/treebanks/tr_imst/index.html, 2024.

[8] U. Türk, F. Atmaca, Ş. B. Özateş, A. Köksal, B. Ozturk Basaran, T. Gungor, and A. Özgür, "Turkish treebanking: Unifying and constructing efforts," in *Proceedings of the 13th Linguistic Annotation Workshop* (A. Friedrich, D. Zeyrek, and J. Hoek, eds.), (Florence, Italy), pp. 166–177, Association for Computational Linguistics, Aug. 2019.

[9] U. Dependencies, "Ud turkish pud." https://universaldependencies.org/treebanks/tr_pud/index.html, 2024.

[10] U. Dependencies, "Universal dependencies turkish comparison treebanks." https://universaldependencies.org/treebanks/tr-comparison.html, 2023.

[11] N. Cesur, A. Kuzgun, M. Kose, and O. T. Yıldız, "Building annotated parallel corpora using the ATIS dataset: Two UD-style treebanks in English and Turkish," in *Proceedings of the 17th Workshop on Building and Using Comparable Corpora (BUCC) @ LREC-COLING 2024* (P. Zweigenbaum, R. Rapp, and S. Sharoff, eds.), (Torino, Italia), pp. 104–110, ELRA and ICCL, May 2024.

[12] Ö. Bakay, Ö. Ergelen, E. Sarmış, S. Yıldırım, B. N. Arıcan, A. Kocabalcıoğlu, M. Özçelik, E. Sanıyar, O. Kuyrukçu, B. Avar, and O. T. Yıldız, "Turkish WordNet KeNet," in *Proceedings of the 11th Global Wordnet Conference* (P. Vossen and C. Fellbaum, eds.), (University of South Africa (UNISA)), pp. 166–174, Global Wordnet Association, Jan. 2021.

[13] B. Marşan, N. Kara, M. Özçelik, B. N. Arıcan, N. Cesur, A. Kuzgun, E. Sanıyar, O. Kuyrukçu, and O. T. Yildiz, "Building the Turkish FrameNet," in *Proceedings of the 11th Global Wordnet Conference* (P. Vossen and C. Fellbaum, eds.), (University of South Africa (UNISA)), pp. 118–125, Global Wordnet Association, Jan. 2021.

[14] A. Kuzgun, N. Cesur, B. N. Arıcan, M. Özçelik, B. Marşan, N. Kara, D. B. Aslan, and O. T. Yıldız, "On building the largest and cross-linguistic turkish dependency corpus," in *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, pp. 1–6, 2020.

[15] N. Kara, B. Marşan, Bakay, A. T. Bayrak, E. Özbek, and O. Yildiz, "Türkçe'nin bağlılık analizi için otomatik bir yaklaşım," in *Türkçe'nin Bağlılık Analizi için Otomatik bir Yaklaşım*, 10 2020.

[16] S. Software, "Starlang software." `https://github.com/starlangsoftware`, 2024.

[17] U. Dependencies, "Ud turkish conllu format." `https://universaldependencies.org/format.html`, 2024.

[18] U. Dependencies, "Universal features." `https://universaldependencies.org/u/feat/index.html`, 2024.