Journal of Theoretical and Applied Information Technology
15th April 2026. Vol.104. No.7

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ENHANCING DEPENDENCY PARSING FOR TELUGU-ENGLISH CODE-MIXED TEXT: TREEBANK CREATION, PARSER ADAPTATIONS AND POS TAGGING INTEGRATION

SANDEEP MADDU[1] , VIZIANANDA ROW SANAPALA[2]

[1,2] Department of Computer Science & Systems Engineering, Andhra University College of Engineering, Andhra University, Visakhapatnam, India.

E-mail: [1]sandeep.maddu@gmail.com, [2]vizianand62@gmail.com

ABSTRACT

Code-mixed text from social media poses significant challenges for syntactic analysis due to irregular grammar, non-standard usage, and frequent language switching. For Telugu-English code-mixed text, the absence of large-scale syntactic resources and specialized parsing models limits progress in downstream multilingual NLP applications. In this work, we address this gap by introducing the first substantial manually annotated Telugu-English code-mixed dependency treebank of 4,152 sentences, developed using Universal Dependencies (UD) 2.0 guidelines. We further propose enhancements to a biaffine dependency parser by incorporating a language-aware head-dependent bias and relation-specific structural weights to better capture cross-lingual syntactic patterns. Our approach improves parsing performance, achieving 75.53% UAS and 61.86% LAS, with consistent gains over a strong baseline. In addition, we demonstrate that integrating dependency-derived syntactic features into a BiLSTM-CRF model improves part-of-speech tagging, achieving a macro-F1 score of 83.73%, with statistically validated gains. We also re-annotate an existing Telugu-English dataset using UD 2.0 to ensure compatibility with modern syntactic frameworks. Overall, this work provides new annotated resources and modeling strategies that advance syntactic processing for Telugu-English code-mixed text, with broader implications for developing robust NLP systems in low-resource and multilingual settings.

Keywords: *Dependency Parsing, Code-Mixing, Telugu, Part-Of-Speech Tagging, Language Aware Head Bias, Dependency Aware Weights*

1. INTRODUCTION

The progress of NLP tools for low-resource languages lags behind resource-rich languages such as English and Chinese. Telugu is a language of the Dravidian family, primarily spoken in the southern Indian states of Andhra Pradesh and Telangana, with more than 90 million native speakers. With its rich morphology, agglutinative nature, and free-word order, Telugu poses a challenge to traditional parsers trained in English and other dominant languages. The complexity is compounded by code-mixing, which is the phenomenon in which speakers fluidly switch between two or more languages in an utterance [1]. Most Telugu speakers alternate between Telugu and English words, and it is also common in social media-generated content such as posts, comments, and videos.

Robust syntactic analysis is essential for higher-level NLP applications. For example, machine translation of code-mixed text requires an accurate understanding of sentence structure, word-level semantics, and grammatical relations between languages. Without reliable dependency parsing and POS tagging, downstream tasks risk generating incoherent or incorrect output.

Dependency parsing is a core NLP task that models the grammatical structure of a sentence by identifying head-dependent relationships between words. The resulting parse tree offers a compact representation of syntactic relations, using labeled arcs (e.g., NSUBJ, OBJ, AUX) to indicate grammatical functions. The Universal Dependencies (UD) framework ([2], [3]) provides a cross-linguistically consistent annotation scheme, currently applied in over 150 languages, making it

particularly suitable for multilingual and code-mixed scenarios.

Dependency trees find applications in several downstream tasks, including machine translation, question answering, Named Entity Recognition (NER), sentiment analysis, information extraction, code-switched sentence creation, and grammatical error correction. Dependency parsing of code-mixed text is necessary to build systems that better understand multilingual conversations, better monitor online hate speech and abusive content, generate code-mixed text that is syntactically correct, and other downstream applications. However, syntactic parsers trained on monolingual Telugu or English text will fare poorly on code-mixed text. The challenge of code-mixed text emanates not just from the use of multiple languages but also the informal nature of usage with no formal rules, non-standard spellings, unpredictable language switches, and lack of quality treebanks.

Telugu's linguistic structure introduces additional complexities. As an agglutinative language with postpositions (as opposed to prepositions), Telugu typically follows a Subject-Object-Verb (SOV) word order and permits a degree of word-order freedom. Telugu dependencies often attach rightward to the head (verb at the end), while English dependencies attach leftward. Functional elements such as auxiliaries and case markers differ significantly from English, affecting syntactic dependencies. Despite its prevalence, Telugu-English code-mixed syntax remains underexplored. To the best of our knowledge, there exists no syntactic treebank or parser for Telugu-English code-mixed text, except for one limited experimental effort on just 300 sentences. In contrast, similar work was undertaken for Hindi-English [4] and Bengali-English [5]. However, no such resource exists for Dravidian code-mixed text, underscoring a notable gap in resources for Dravidian language pairs.

POS tagging is another fundamental task for many NLP applications. However, annotated datasets for POS tagging of Telugu-English code-mixed text are scarce. The existing works are mostly based on the ICON 2016 shared task, which uses Google's Universal POS tagset [6]. Besides creating a new dataset that addresses the shortage of annotated resources, we show that incorporating syntactic features derived from dependency trees, such as dependency heads and relation labels, enhances POS tagging performance over models relying solely on lexical and contextual features.

The primary contributions of this paper are as follows:

- Treebank Creation: We present the first substantial manually annotated Telugu-English code-mixed dependency treebank consisting of 4,152 sentences, annotated in accordance with the Universal Dependencies 2.0 guidelines.
- Parsing Enhancements: We enhance a state-of-the-art biaffine dependency parser by introducing (i) a language-conditioned head-dependent bias matrix and (ii) dependency relation-specific structural weights. These modifications help the parser better model syntactic transitions in code-mixed text.
- POS Tagging Integration: We integrate dependency-based syntactic features into a BiLSTM--CRF POS tagger for Telugu-English code-mixed text and re-annotate the ICON 2016 dataset with UD 2.0 for compatibility. Using a controlled ablation with paired-bootstrap testing, we show that the core features (HEAD and DEPREL) alone give only a small gain, while the additional structural features provide complementary signal beyond them.
- All the resources, including treebank and re-tagged ICON data, are publicly released for benefit of research community.

Beyond the development of resources and models, this study contributes to knowledge creation in the domain of code-mixed and multilingual NLP by providing new empirical and methodological insights that were previously lacking for Telugu-English code-mixed text. First, the manually annotated dependency treebank enables systematic analysis of cross-lingual syntactic structures in informal social media data, offering evidence of how grammatical relations manifest across language boundaries in a low-resource setting. Such insights were not possible due to the absence of high-quality syntactic datasets for this language pair.

Second, the proposed language-aware head-dependent bias introduces a principled mechanism for modeling cross-lingual head selection, providing empirical validation that language identity plays a critical role in determining syntactic structure in code-mixed text. This extends prior work, which largely treats multilingual inputs without explicitly encoding language-conditioned structural preferences.

Third, the use of relation-specific structural weighting demonstrates that dependency relations vary in their contribution to parsing accuracy, highlighting the importance of structurally central relations in code-mixed syntactic modeling. This offers a new perspective on how syntactic importance can be incorporated into learning objectives for parsing.

Finally, the integration of dependency-derived structural features into POS tagging provides empirical evidence that syntactic structure contributes meaningfully to sequence labeling in code-mixed contexts, particularly when combined with positional and hierarchical information. This finding refines existing assumptions in the literature, which often report limited gains from dependency features alone.

The significance of these contributions lies in advancing both the theoretical understanding and practical modeling of code-mixed syntax. By addressing the lack of annotated resources, introducing language-aware and structurally informed modeling strategies, and demonstrating their effectiveness across tasks, this work establishes a foundation for future research in low-resource multilingual NLP and provides a generalizable framework for other code-mixed language pairs.

2. LITERATURE REVIEW

2.1 Neural Approaches to Dependency Parsing

Modern dependency parsers broadly fall into transition-based and graph-based categories. Transition-based parsers [7] incrementally build trees using shift-reduce operations, where arcs are formed based on the stack configuration and prior actions. In contrast, graph-based parsers such as [8] construct fully connected graphs and score each potential dependency arc to select the optimal tree.

The introduction of deep learning significantly advanced both paradigms. One of the earliest neural models [9] used feed-forward networks with hand-crafted features for transition-based parsing. Later, [10] introduced BiLSTM encoders combined with MLP-based arc scoring, enabling better context modeling. A significant leap came with the Deep Biaffine Attention Parser [11], which employs BiLSTMs to generate contextual representations followed by biaffine transformations for scoring arcs and dependency labels. This model has become the foundation for many parsers. Extensions such as UDify [12] replaced BiLSTM encoders with BERT [13], enabling joint multilingual training. Recent works on use of LLMs in dependency parsing [14-15] explores the use of large language models for generating Universal Dependencies annotations in code-switched text, demonstrating that prompt-based LLM pipelines combined with expert validation can effectively bootstrap syntactic resources for low-resource language pairs. The findings also highlight that traditional parsers trained on monolingual treebanks fail to generalize well to mixed-language inputs, reinforcing the need for dedicated resources and specialized parsing strategies for code-mixed languages. The findings further indicate that current LLMs still struggle with robust comprehension of code-switched text, underscoring the need for specialized resources and models for code-mixed syntactic analysis.

2.2 Dependency Parsing in Code-Mixed Indian Languages

Code-mixed parsing, especially for Indian social media text, presents unique challenges, including inconsistent grammar and informal spellings. While Hindi-English and Bengali-English have seen some progress, Telugu-English remains underexplored. [4] proposed a neural stacking model based on [16] for Hindi-English that leveraged syntactic knowledge from monolingual Hindi and English treebanks using a shared POS tagging layer. [5] extended this idea to Bengali-English using synthetic code-mixed data. [17] added auxiliary semi-supervised tasks to improve parsing across multiple language pairs.

Despite the prevalence of Telugu-English code mixing, resources on dependency parsing remain limited. [18] annotated 300 Telugu-English sentences and reported low LAS (11.79%). No substantial public dataset or comprehensive parser exists for Telugu-English code-mixing. Further, available POS-tagged corpora use incompatible tagsets (e.g., Google Universal POS tags in ICON 2016), due to which direct adaptation of joint modeling architectures such as Stanza and UDify, that use UD style annotations, is not feasible. By creating the first substantial Telugu-English UD treebank and re-annotating POS data with a consistent UD based tagset, our work provides the necessary foundation for exploring joint modeling in future research.

2.3 POS Tagging in Code-Mixed Settings

Early approaches in POS tagging of code-mixed text followed language-specific pipelines, where words were tagged using respective monolingual taggers and labels were mapped to a shared tagset [19]. For instance, [20] used this approach for

Hindi-English; [21] applied it to Bengali-English. These methods, however, ignored sequence-level dependencies that can disambiguate POS labels.

Later models leveraged statistical and deep learning techniques. [22] used CRF, BiLSTM, and BiLSTM-CRF models for Kannada-English. For Telugu-English, most work focused on the ICON 2016 shared task [23], using Google's Universal POS tagset. Systems used CRFs [24], Decision Trees, Random Forests [25], and Deep Learning (RNNs [26], BiLSTM-CRF [19], Transformers [27]). Joint models for POS and language identification were also explored [26].

The best reported accuracy for Telugu-English POS tagging is 91.65% [28], with best F1 score of 87.9% from [29]. Recent work shows that POS tagging and dependency parsing are mutually beneficial. [30] proposed a multi-task model jointly optimizing POS, parsing, and other tasks. [31] observed that POS tagging and parsing are correlated tasks. Despite these advances, most Telugu-English POS taggers operate independently of syntactic structure, leaving unexplored the potential benefits of dependency-informed tagging.

2.4 Research Gaps

Despite significant progress in dependency parsing and code-mixed NLP, several critical gaps remain unaddressed, particularly for Telugu-English code-mixed text. First, existing work on dependency parsing for code-mixed languages has largely focused on Hindi-English and Bengali-English, with Telugu-English receiving minimal attention. The only available efforts are limited in scale and do not provide sufficiently large or diverse annotated datasets for robust model training and evaluation.

Second, most dependency parsing approaches rely on monolingual or multilingual models that do not explicitly account for language identity in syntactic decision-making. As a result, they fail to capture cross-lingual structural variations inherent in code-mixed text, such as differences in word order and head-directionality between Telugu and English.

Third, recent advances using large language models attempt to address resource scarcity through automatic annotation; however, these approaches depend on the availability of reliable gold-standard datasets and exhibit limitations in accurately modeling syntactic structures in mixed-language inputs. This highlights the continued need for high-quality manually annotated resources.

Fourth, existing POS tagging approaches for Telugu-English code-mixed text are typically developed independently of syntactic parsing and rely primarily on lexical and contextual features. This overlooks the potential contribution of dependency-based structural information for resolving ambiguity in code-mixed settings.

These gaps collectively indicate the need for (i) a large-scale, high-quality dependency treebank for Telugu-English code-mixed text, (ii) parsing models that incorporate language-aware and structurally informed mechanisms, and (iii) integrated approaches that leverage syntactic information to improve downstream tasks. The present work addresses these gaps through the creation of a UD-based treebank, enhancements to a biaffine dependency parser, and the integration of dependency-derived features into POS tagging.

Problem Statement: Despite recent advances in multilingual and code-switched NLP, existing approaches exhibit several critical limitations when applied to Telugu-English code-mixed text. First, there is a lack of large-scale, high-quality dependency treebanks for this language pair, which restricts the training and evaluation of syntactic models. Second, prior dependency parsers are predominantly trained on monolingual or weakly code-mixed data and fail to capture the structural variability and cross-lingual interactions inherent in informal social media text. Recent LLM-based approaches attempt to address resource scarcity through automatic annotation; however, they still depend on reliable gold-standard datasets and exhibit limitations in handling syntactic structures in mixed-language inputs. Third, most existing POS tagging models for Telugu-English operate independently of syntactic information, overlooking the potential benefits of dependency-based structural cues for disambiguation. These limitations highlight the need for (i) a robust, manually annotated dependency treebank tailored to Telugu-English code-mixed text, (ii) parsing models that explicitly incorporate language-aware and structurally informed mechanisms, and (iii) integrated approaches that leverage syntactic information to improve downstream tasks such as POS tagging.

The limitations identified in prior work not only indicate a technical gap but also constrain knowledge creation in the domain of multilingual and code-mixed NLP. In particular, the absence of high-quality syntactic resources for Telugu-English code-mixed text restricts the development of reliable parsing models and limits the understanding of cross-lingual syntactic interactions in informal communication. Existing approaches, including recent LLM-based methods, primarily rely on weak

supervision or automatic annotation and therefore lack the linguistic reliability required for advancing syntactic theory and model interpretability in code-mixed settings. This study addresses these gaps by contributing both resources and methodological advancements. First, it enables knowledge creation through the development of a manually annotated dependency treebank that captures real-world syntactic patterns in Telugu-English code-mixed text. Second, it introduces language-aware and structurally informed parsing mechanisms that model cross-lingual dependencies more effectively. Third, it demonstrates how syntactic information can be systematically integrated into downstream tasks such as POS tagging, thereby providing empirical evidence for the role of structure in improving performance. Together, these contributions establish a foundation for future research in code-mixed syntactic analysis and support the development of robust multilingual NLP systems for low-resource languages.

Based on the above gaps, this study is guided by the following research questions:

- How can a high-quality dependency treebank for Telugu-English code-mixed text be constructed to support reliable syntactic analysis?
- To what extent can incorporating language-aware head-dependent bias improve dependency parsing performance in code-mixed text?
- How do dependency relation-specific structural weights influence parsing accuracy, particularly for structurally important relations?
- Can dependency-derived syntactic features improve POS tagging performance in Telugu-English code-mixed text, and which features contribute most significantly?
- How do the proposed methods compare with existing approaches in handling structural variability and cross-lingual interactions in code-mixed text?

3. DATA

We present the TEMPLE corpus (Telugu-English Mixed Parse Labelled Examples), a manually annotated dependency treebank for Telugu-English code-mixed text developed in compliance with the Universal Dependencies (UD) 2.0 framework. We adopt the UD framework due to its multilingual coverage and cross-linguistically consistent inventory of dependency relations. UD allows for seamless integration of Telugu and English syntactic structures within a single treebank, avoiding language-specific label incompatibilities.

3.1 Corpus Source and Sentence Selection

The source corpus comprises 3.6 million Telugu-English code-mixed YouTube comments collected by [32]. The comments cover various domains, including politics, films, sports, entertainment, and general. To ensure the selection of linguistically rich and syntactically informative content, we filtered this corpus to retain sentences containing 10--20 tokens. This range excludes extremely short sentences that lack syntactic complexity and very long ones that introduce multi-clausal structures and annotation ambiguity. From this subset, we randomly sampled 4,152 sentences and manually annotated them for POS tags and syntactic dependencies.

3.2 Annotation Methodology

Annotation was performed using the Prodigy tool, with outputs in the CoNLL-U format (https://universaldependencies.org/format.html). Each token is annotated with ten UD-specified fields: ID, FORM, LEMMA (field defined in UD format but not used in this work), UPOS, XPOS (not used), FEATS (not used), HEAD, DEPREL, DEPS (not used), and MISC. A sample annotation is shown in Table 1 and a sample dependency tree is shown in Figure 1.

Due to the high cognitive demands of dependency annotation and the scarcity of trained Telugu-English bilingual annotators familiar with the UD framework, all annotations were performed by a single annotator (first author). While this precludes the computation of inter-annotator agreement scores, we ensured annotation consistency via multiple rounds of self-review and cross-validation with UD guidelines. We have made the annotated dataset publicly available for community validation and refinement. Given the scarcity of gold-standard syntactic resources in this domain, our annotations serve as an initial, manually verified resource for training and benchmarking.

3.3 Corpus Statistics

The annotated corpus contains 51,669 tokens across 4,152 sentences. We randomly split the annotated dataset into 70% train, 10% dev, and 20% test (Table 2). All evaluations reported in this paper are conducted on the held-out test set.

Table 3 summarizes the frequency of various dependency relations in the TEMPLE corpus. The most frequent relation is CASE (8.62%), which is expected given the agglutinative nature of Telugu and the frequent use of postpositions and case markers. Core syntactic relations such as NSUBJ (8.51%) and OBJ (7.92%) occur with high frequency, reflecting the prominence of subject–object-verb constructions. The ROOT relation accounts for 8.13% of tokens, aligning with the expectation that each sentence contributes one root. PARATAXIS (6.78%) is high, due to frequent use of loosely connected clauses, common in conversational and informal speech where speakers switch topics or insert commentary, a trait observed more in code-mixed contexts.

Modifiers such as NMOD (6.49%), OBL (6.02%), and ADVMOD (5.72%) are also well represented, indicating rich use of noun and adverbial phrases. The relatively high count of VOCATIVE (6.20%) reflects the discourse-oriented nature of social media language, where direct address (e.g., "bro," "anna") is common. The presence of DISCOURSE (3.54%) further supports this observation, indicating the frequent use of interjections.

Other functional categories like AUX (3.16%) and AMOD (4.78%) contribute to expressing verbal aspect and modifying noun phrases, respectively. Clausal and complement relations such as ADVCL (2.74%), XCOMP (1.24%), ACL (1.05%), CCOMP (0.68%), and CSUBJ (0.23%) appear less frequently but demonstrate the syntactic complexity present in the data. DET (2.20%) and NUMMOD (1.28%) support the construction of noun phrases. Coordinating (CC: 1.05%) and subordinating (MARK: 0.86%) conjunctions indicate presence of both coordination and subordination structures. FLAT (1.63%) which refers to multi-word names and iconic sequences, and COMPOUND (2.01%) which is a combination of words that behave as single word, such as "face value" are also represented.

Less frequent tags such as FIXED (0.45%), GOESWITH (0.61%), and APPOS (0.45%) represent multiword expressions and appositional phrases. Rare relations like DEP (0.04%), EXPL (0.01%), REPARANDUM, and ORPHAN (nearly 0%) are marginal. Overall, the distribution reflects a syntactically diverse and conversationally rich code-mixed corpus with varied syntactic constructions.

The distribution of Universal POS tags (Table 4) shows prevalence of NOUN (28.11%), VERB (20.25%), and ADP (7.79%), aligned with nominal richness and complex verb-noun structures of Telugu-English text. The corpus is lexically rich and reflects high discourse variability.

3.4 Reannotation of ICON 2016 Dataset

Almost all the existing works on POS tagging of Telugu-English code-mixed text use the ICON 2016 Telugu-English POS tagging dataset [23]. The original version used Google's Universal POS tagset [6]. However, we observed several inconsistencies and annotation errors in the original dataset, especially with overuse of the placeholder tag G_X, which occurred in 6,626 instances (22.48% of tokens). Following a detailed analysis, we chose to re-annotate the entire corpus from scratch. To support comparative evaluation, we re-annotated the ICON dataset using UD 2.0 POS tagset. UD tags are designed to be fully compatible with UD dependency relations. Since our work directly integrates POS tags into dependency parsing, using a tagset that aligns with UD's dependency label set ensures cross-task consistency, which is impossible with Google's tagset without custom mapping. Also, UD 2.0 has a large, active community and treebanks for over 150 languages, including Indian languages. This facilitates future cross-lingual transfer and comparative evaluation. Our re-annotated version contains 1,980 sentences and 29,471 tokens, split as shown in Table 5.

Table 6 shows POS distribution in the re-annotated ICON data. Compared to TEMPLE, it includes more PROPN, SYM, and PUNCT, reflecting its origin from informal platforms such as WhatsApp and Facebook. TEMPLE, in contrast, was curated for syntactic completeness and reduced noise.

3.5 Dataset Availability

We publicly release both the TEMPLE corpus and the re-annotated ICON dataset to support future research in low-resource and code-mixed NLP.

4. METHODOLOGY

This section describes the architecture and enhancements made to the dependency parsing framework for Telugu-English code-mixed text, as well as the integration of syntactic features into a POS tagging pipeline.

4.1 Base Dependency Parser

Our dependency parser builds upon the deep biaffine attention model by ([11], [33], [34]). We

use the Stanza toolkit [35], which implements this parser. The word, POS, and character embeddings are concatenated with contextual embeddings from the transformer model. We adopt XLM-RoBERTa [36] as the multilingual encoder and extract token representations from its top 4 hidden layers. These embeddings are then passed to a BiLSTM encoder. The parser uses biaffine classifiers to score dependency arcs and their labels.

We build on Stanza's graph-based biaffine parser rather than UDify because UDify's multilingual, multitask design is optimized for cross-lingual transfer across dozens of UD treebanks, whereas our focus is on high-quality parsing of a single code-mixed treebank. This allows us to allocate model capacity entirely to dependency parsing and code-mixing–specific enhancements rather than multitask learning.

4.2 Language-Aware Head-Dependent Bias (LAHB)

To account for syntactic divergences between Telugu (head-final, SOV) and English (head-initial, SVO), we introduce a learnable Language-Aware Head-Dependent Bias (LAHB). This bias helps encode cross-lingual head direction preferences using a trainable matrix. Its bias allows the parser to prefer certain cross-lingual head directions based on training data, e.g., that Telugu adjectives more often modify Telugu nouns than English ones, or that English verbs rarely take Telugu auxiliaries. The mechanism works by injecting soft, learnable biases into the arc scoring function of the parser.

Specifically, we define a learnable bias matrix:

$$\text{lang\_head\_bias} \in \mathbb{R}^{L \times L},$$

where L is the number of language tags (Telugu, English, Named Entity, Other). Each entry lang_head_bias[i, j] encodes the preference of a token in language i acting as the head of a token in language j. A global scaling parameter

$$\text{bias\_scale} \in \mathbb{R}$$

controls the contribution of these biases.

Formally, for each potential head–dependent pair (h, d), the arc score is updated as:

$$\text{unlabeled\_scores}_{h,d} \leftarrow \text{unlabeled\_scores}_{h,d} + \text{bias\_scale} \cdot \text{lang\_head\_bias}[\ell(h), \ell(d)],$$

where $\ell(h)$ and $\ell(d)$ denote the language tags of the head and dependent tokens, respectively.

The learned biases are trained jointly with the parser. The above modification is shown in Figure 2.

We apply the language-aware head-dependent bias only to the unlabeled scores and not to the dependency relation scores. This decision is based on the intuition that language identity influences the likelihood of syntactic head selection more directly than the choice of dependency relation label. For instance, word order differences between Telugu and English can shift likely head positions, making cross-language head bias useful for head prediction. In contrast, once a head is selected, the dependency label is more influenced by local syntactic and semantic context. Additionally, injecting language bias into both arc and label components adds more parameters and complexity, potentially leading to overfitting, especially in low-resource code-mixed settings. Therefore, we apply the language-aware head-dependent bias only to the unlabeled scores.

We preserve the architectural separation between arc prediction and relation labeling as in [31]. The language-aware head–dependent bias is applied only to the arc scorer's unlabeled scores, which are optimized using a cross-entropy loss. Relation labeling uses a separate score computed by the biaffine label classifier, optimized with its own cross-entropy loss. The total parsing loss is the sum of these two losses. These modules have distinct parameter sets, and no gradient flow occurs between arc scoring and relation labeling except through shared BiLSTM encodings, ensuring that the two tasks remain independent.

4.3 Dependency Relation-Specific Loss Weighting

To reflect the structural importance of different dependency labels in code-mixed Telugu-English parsing, we assign each relation r a scalar weight w_r, defined as:

$$w_r = \alpha \cdot \text{centrality}_r + \beta \cdot \text{frequency}_r \cdot \text{sensitivity}_r$$

where α and β are hyperparameters that balance the contribution of structural centrality versus frequency-based sensitivity.

We consider $\alpha = 0.70$ and $\beta = 0.30$. The coefficients are fixed based on linguistic and structural considerations, prioritizing interpretability and replicability over data-driven optimization. The higher emphasis on the centrality term (α) reflects the intuition that, in code-mixed parsing, structurally pivotal relations such as ROOT, NSUBJ, and OBJ have a large impact on syntactic coherence. The lower weight for the frequency–sensitivity component (β) ensures structurally peripheral relations, even if frequently occurring, do not dominate the scoring, while still allowing label frequency and observed error sensitivity to influence the weights.

Centrality: We construct a directed graph over dependency relations, linking parent and child labels in parse trees. Each node's centrality is computed as the mean of its degree, betweenness, and closeness centrality:

$$\text{centrality}_r = \frac{1}{3}\left(C_r^{\text{deg}} + C_r^{\text{btw}} + C_r^{\text{cls}}\right)$$

This highlights structurally central labels that frequently act as syntactic central points.

Frequency: We calculate the relative token-level frequency of each relation across training data:

$$\text{frequency}_r = \frac{\text{count}(r)}{\sum_{r'} \text{count}(r')}$$

This helps down-weight overrepresented, low-impact labels like CASE and PUNCT.

Sensitivity: Sensitivity quantifies the drop in LAS when a given relation is removed. Values are normalized and log-scaled to prevent distortion from outliers.

Normalization: Final weights are min-max normalized for stability:

$$w_r \leftarrow \frac{w_r - \min(w)}{\max(w) - \min(w)}$$

While we experimented with making these weights learnable, the results did not yield improvements. This may be due to the limited size of our dataset (4,152 sentences; 51,669 tokens), which may not provide sufficient signal for the model to reliably learn 37 independent weight parameters. Moreover, allowing the model to learn relation-specific biases directly could lead to overfitting or instability in low-resource settings. In contrast, our handcrafted weighting scheme grounded in empirical distributional and structural statistics offered more stable improvement. Nonetheless, with larger and more diverse code-mixed corpora, learnable relation weights could be a promising direction for future work.

These modifications are integrated into the parser's neural architecture by editing core modules in the Stanza codebase. All models are trained using Google Colab, under a PyTorch-based implementation. We retain Stanza's default settings for most hyperparameters, with a few adjustments where necessary. The key configuration values are listed below.

4.4 Training Details

We trained and evaluated all models using Google Colab with NVIDIA L4 GPU (24 GB VRAM) with PyTorch 2.6.0, Transformers 4.53.2, Tokenizers 0.21.2, Stanza 1.10.1, and SentencePiece 0.2.0. The hyperparameters are summarized in Table 7.

4.5 POS Tagging with Dependency Features

Our POS tagging feature set was designed to capture the lexical, contextual, language-specific, and syntactic cues. Lexical features capture surface clues for POS tags, like suffixes for Telugu verbs and nouns or case and capitalization for English proper nouns. Context features (two words before and after) capture short-range patterns, while language tag features help the POS tagger follow language-specific rules. Syntactic features integrate structural information from the dependency parser. We use these features in a BiLSTM-CRF model [37], implemented using NCRF++ [38].

Features:

- Lexical: word length, special characters, case features, digits, prefixes/suffixes, presence in Telugu dictionary [39], TF-IDF, Soundex encoding [40], normalized form [36].
- Context: two preceding and following words.
- Language tag: predicted from a separate LID model [32].
- Syntactic: dependency head, relation, head POS, relative position, tree depth, subtree size, root indicator. We ensure that all syntactic features come from predicted parses, preventing data leakage.

The syntactic features capture structural and relational properties from the dependency parse of a sentence. They provide information about how words are connected and organized.

(1) Dependency Head: Refers to the governing word (head) of the current token in the dependency tree. This captures hierarchical relations and grammatical structure.
(2) Relation (Dependency Relation): The type of grammatical relation between the token and its head. These relations encode how words function in a sentence.
(3) Head POS: The part-of-speech tag of the dependency head. This provides additional context: for example, if the head is a verb, its dependents are likely to be subjects, objects, or modifiers.
(4) Relative Position: Indicates whether the head is to the left or right of the current token. For example, in SVO (Subject-Verb-Object) structures like English, subjects usually appear to the left of the verb, while in SOV languages like Telugu, objects precede the verb. This is very useful for code-mixed text, where word-order patterns may shift.

(5) Tree Depth: The distance of the token from the root in the dependency tree. Shallow depth often corresponds to core sentence elements (subject, predicate), while deeper depth corresponds to modifiers or nested structures.
(6) Subtree Size: The number of tokens in the subtree rooted at the current token. This helps capture whether the token governs a large phrase (e.g., a verb with multiple dependents) or just a small unit.
(7) Root Indicator: A binary feature that marks whether the token is the root of the dependency tree (usually the main predicate of the sentence). The root carries global sentence-level importance.

Together, these features provide both local syntactic cues (head, relation, POS, relative position) and global structural cues (depth, subtree size, root), making them informative for parsing and downstream NLP tasks, especially in code-mixed text where syntax can be non-standard.

The POS tagging model is trained on Google Colab using T4 GPU for 25 epochs with learning rate 0.015, dropout 0.5, and L2 regularization 1e-8.

5. RESULTS

5.1 Dependency Parsing Performance

We evaluate our dependency parser using standard metrics: Unlabeled Attachment Score (UAS) and Labeled Attachment Score (LAS), which measure the percentage of words correctly attached to their syntactic heads, with or without considering dependency labels.

The reported headline results for UAS/LAS and POS tagging are averaged over five independent runs with seeds {42, 1234, 2025, 7, 31415}. All other experiments such as ablations, significance testing are conducted using a single model run (seed~42) for efficiency; the base condition in the experiments therefore reflects that run's baseline performance (76.12~UAS / 63.04~LAS / 84.02~POS F-score) rather than the five-seed average reported in the headline results (75.53~UAS / 61.86~LAS / 83.73~POS F-score). Performance difference (Δ)~values are computed relative to the seed~42 baseline.

Our proposed model, enhanced with language-aware head-dependent bias and dependency relation-specific weights, achieves an average UAS of 75.53 ± 0.71 (mean ± standard deviation) and LAS of 61.86 ± 1.08. While UAS variation across seeds is under 1, LAS variation is slightly higher due to the increased difficulty of correctly identifying both heads and labels in Telugu-English code-mixed text. The consistent ranking of our model over the baseline across all seeds indicates that improvements are robust despite natural stochasticity in training.

Table 8 presents a comparison of our results with prior research on dependency parsing for various code-mixed language pairs. To the best of our knowledge, this work constitutes the first substantial study on Telugu-English dependency parsing. The only previous effort in this direction is by [18], whose study is limited in scope, with a relatively small dataset of 300 sentences and modest performance (30.61 LAS, 11.79 UAS). Consequently, no directly comparable work exists for Telugu-English. For completeness, we report performance measures on other code-mixed language pairs in Table 8. Nevertheless, these results should be interpreted with caution, as the underlying corpora differ significantly in size, domain, and linguistic characteristics, rendering direct comparisons across language pairs neither meaningful nor conclusive. This underscores the novelty and significance of our work, which establishes the first meaningful benchmark for dependency parsing in Telugu-English code-mixed text, thereby providing a valuable foundation for future research and advancements in this area.

5.2 Ablation Study on Dependency Parsing

We conduct an ablation study to assess the contribution of each enhancement. Table 9 reports UAS and LAS scores under different settings. Both enhancements contribute independently to performance, with weights having a slightly greater impact than head bias.

- Removing relation weights results in a 2.00% drop in LAS.
- Removing head-language bias results in a 1.30% drop in LAS.
- Removing both results in a 1.73% drop in LAS

5.3 POS Tagging Performance

We evaluate our dependency parser using standard metrics: accuracy, precision, recall, macro F1 score. On the TEMPLE dataset, the model achieves accuracy of 86.24% and macro F1 score of 78.16%. However, since most of the existing works on POS tagging of Telugu-English code-mixed text use the ICON 2016 dataset, we report our performance on this dataset also.

To ensure a fairer comparison with the TEMPLE dataset which is UD style annotated, the ICON data was fully re-annotated to the UD 2.0 tagset, thereby aligning label definitions and reducing

inconsistencies. This re-annotation addresses known issues in the original ICON Universal POS tagset, particularly systematic mislabeling of G_X tokens—while maintaining the original sentence content and domain characteristics.

POS tagging results on the ICON (UD tagset) dataset (Table 10) achieve an accuracy of 91.76 ± 0.06% and an F1-score of 83.73 ± 0.24%. This is higher than both [19] (F1: 73.55%) and [28] (F1: 61.52%), the only prior works that used the corrected ICON dataset. Since both the re-annotated ICON dataset and TEMPLE dataset use the same UD tagset, the performance gap between TEMPLE (86.24% accuracy, 78.16% F1) and ICON - UD (91.76% accuracy, 83.73% F1) is not attributable to tagset mismatch, but to intrinsic dataset differences such as domain, sentence length, and code-mixing patterns.

For completeness, when our model is applied to the ICON dataset in its original Universal POS tagset form, accuracy drops sharply to 69.93% and F1 to 67.51%. This may be attributed to both tagset difference and annotation errors (mislabeling of G_X tokens in the original ICON Universal POS dataset).

5.4 Ablation Study on POS Tagging

Table 11 shows the drop in F1 when specific feature groups are removed. All ablations are performed by removing one feature group at a time from the complete feature set and retraining the model from scratch, keeping all other features intact.

Lexical features cause the most significant drop (−1.47), followed by syntactic head features (−1.10) and language identification tags (−0.99). Contextual features have minimal impact, suggesting that BiLSTMs already capture adequate sequential dependencies.

Dependency features provide syntactic disambiguation that lexical features miss, especially in code-mixed settings where syntax is non-standard. Removing both head and dependency relation features leads to a 1.38% drop in F1, signifying that syntactic context from dependency parsing contributes strongly to POS performance. "head-related" features (head + headpos) cause a 1.10% drop, and "structural tree" features (depth, subtree, isroot) result in a 0.97% decline. This shows that both the head information and its position within the dependency tree are valuable. Removing only dependency relations causes a minimal drop of 0.12%, suggesting that the relation type is less informative in isolation than in combination with the head.

The above ablation study suggests that both lexical and dependency-based features are important and that feature design should prioritize a combination of syntax (dependency tree), language cues (dictionaries, language ID) and morphology. Their synergy enables robust POS tagging in noisy, informal code-mixed data.

5.5 Discussion in Context of Research Problem and Evaluation Criteria

The evaluation of the proposed approach is grounded in standard metrics for syntactic analysis, namely Unlabeled Attachment Score (UAS) and Labeled Attachment Score (LAS) for dependency parsing, and macro-F1 for POS tagging. These metrics are widely used in prior dependency parsing and sequence labeling studies, ensuring comparability with existing work. UAS measures the correctness of head selection, while LAS evaluates both head and relation labeling, making them particularly suitable for assessing syntactic structure in code-mixed text, where both attachment and relation assignment are challenging due to language switching. Macro-F1 is chosen for POS tagging to account for class imbalance and to ensure that performance on less frequent tags is not overshadowed by dominant categories.

The choice of these evaluation criteria is significant in the context of the research problem, as code-mixed text introduces variability in word order, morphology, and syntactic patterns across languages. Improvements in UAS indicate better modeling of cross-lingual head-dependent relationships, while gains in LAS reflect enhanced ability to assign appropriate grammatical roles in mixed-language contexts. Similarly, improvements in macro-F1 demonstrate that incorporating syntactic features helps resolve ambiguity in POS tagging, particularly for structurally complex or low-frequency categories.

These evaluation practices are consistent with prior studies in dependency parsing and code-mixed NLP, where UAS and LAS serve as the primary metrics for syntactic accuracy, and F1-based measures are used for POS tagging. However, unlike many previous works that rely solely on overall scores, our analysis extends to relation-wise performance, ablation studies, and statistical significance testing, providing a more comprehensive understanding of model behavior.

The observed improvements in parsing performance can be directly linked to the proposed model enhancements. The language-aware head-dependent bias improves head selection in cross-lingual contexts, which is reflected in increased

UAS. The relation-specific weighting mechanism enhances the prediction of structurally important relations, contributing to LAS improvements. These findings align with prior observations that code-mixed parsing requires explicit modeling of cross-lingual interactions, which are not captured by monolingual or standard multilingual parsers.

Similarly, the improvements in POS tagging performance support the hypothesis that syntactic information provides complementary signals beyond lexical and contextual features. While previous studies have shown limited gains from dependency features alone, our results demonstrate that combining head information with structural features such as tree depth, relative position, and subtree size yields statistically significant improvements. This suggests that POS disambiguation in code-mixed text benefits from both local syntactic cues and global structural context.

When considered alongside existing literature, our findings reinforce known challenges in code-mixed NLP, such as ambiguity in argument structure, variability in word order, and limitations of models trained on monolingual data. At the same time, our results extend prior work by demonstrating that language-aware modeling and structurally informed features can effectively address these challenges. Overall, the study provides empirical evidence that combining resource creation with targeted model adaptations leads to measurable improvements in syntactic analysis for code-mixed text, thereby directly addressing the research problem identified in this work.

6. DISCUSSION

6.1 Statistical Significance Analysis

To determine whether the observed improvements in Unlabeled Attachment Score (UAS) and Labeled Attachment Score (LAS) are due to systematic enhancements rather than random variation, we conducted a paired bootstrap resampling test [39]. This method is well-suited for dependency parsing evaluation because it respects the paired nature of the experiment, where both systems are evaluated on the same set of sentences.

We performed B=10,000 bootstrap replicates over sentences from the test set. In each replicate, N sentences (where N is the test set size) are sampled with replacement, and UAS/LAS is computed for both the baseline and enhanced systems. The difference (Δ) in scores between systems is recorded for each replicate, and a 95% percentile confidence interval (CI) is computed. We report one-sided p-values for the hypothesis that the enhanced system outperforms the baseline.

The results in Table 12 show that the 95% confidence intervals for both UAS and LAS improvements exclude zero and that $p < 0.001$ in all cases. This confirms that our enhancements lead to statistically significant gains.

6.2 Per-Relation Impact of the parser enhancements

Table 13 presents the LAS performance for each dependency relation, comparing the full model (with head-language bias and dependency-relation-specific weights) against the variant without bias and weights.

Largest positive gains (statistically significant, $p < 0.01$).

- **NMOD** (+22.7 pp, $p < 0.0001$, CI [+0.19, +0.26]): The largest improvement, reflecting nominal modifiers that often occur across languages (e.g., English noun with Telugu modifier or vice versa). Head-language bias disambiguates head selection in such contexts.
- **NUMMOD** (+8.8 pp, $p = 0.0066$, CI [+0.03, +0.14]): Numeric modifiers benefit when mixed-language pat- terns (e.g., Telugu quantifier + English noun) occur.
- **DET** (+7.27 pp, $p = 0.0259$, CI [+0.01, +0.14]): Determiners more reliably attach to the nouns they modify when the model learns that English determiners tend to modify English nouns, and Telugu equivalents modify Telugu nouns.
- **NSUBJ** (+5.36 pp, $p = 0.0001$, CI [+0.03, +0.08]): Nominal subjects often cross language boundaries in code-mixed sentences, benefiting from language-aware head selection.
- **AMOD** (+4.84 pp, $p = 0.0005$, CI [+0.02, +0.08]): Adjectival modifiers exhibit gains similar to NMOD, particularly in mixed-language noun phrases.

Moderate gains (non-significant). Relations such as **MARK**, **GOESWITH**, and **ADVCL** show moderate LAS improvements (4–8 pp) but without statistical significance due to smaller sample sizes or greater variability.

No meaningful change. High-frequency, monolingual-stable relations like **CASE**, **ADVMOD**, and **VOCATIVE** change by less than 1 pp, indicating that these are already well-predicted without bias.

Negative impact. A few relations see statistically significant declines: **OBL** (–2.89 pp, p = 0.0226), **OBJ** (–3.39 pp, p = 0.0112), **AUX** (–4.5 pp, p = 0.0026), and especially **FLAT** (–37.25 pp, $p < 0.0001$). For OBL, OBJ, and AUX, the bias may over-prioritize language-specific attachment, leading to errors when both tokens share the same language. FLAT suffers because such multiword expressions are typically language-homogeneous, making language bias less helpful.

Zero-score relations. Many infrequent relations (e.g., IOBJ, APPOS) have zero LAS in both models due to extreme data sparsity.

Interpretation. The most significant benefits are observed for NMOD, NUMMOD, DET, NSUBJ, and AMOD—relations that frequently cross language boundaries in Telugu–English code-mixed text. Relations that are mostly monolingual and have strong morphological or positional cues show negligible changes. In contrast, some high-frequency but less language-sensitive relations, as well as language-homogeneous structures, are negatively impacted.

6.3 Do dependency features help POS tagging? A paired bootstrap analysis

We compare three POS taggers on the same test set: (1) a model with no dependency features (lexical/context only), (2) a model with only HEAD+DEPREL, and (3) a model with all dependency features (HEAD, DEPREL, head POS, relative position, tree depth, subtree size, root indicator). We evaluate macro-F1 and test for significance using paired bootstrap resampling over sentences (B=10,000). In each replicate we sample N sentences with replacement (where N is the test size), recompute macro-F1 for both systems being compared, and record the difference Δ. We report the observed Δ, a ,95% percentile confidence interval (CI), and a one-sided p-value for the alternative H1: enhanced > baseline (Table 14).

Findings:

(1) Adding the full set of dependency features to a dependency-free baseline yields a ,+0.88 macro-F1 improvement; the 95% CI includes zero \,[-0.40, 2.20], and the one-sided p=0.0969, i.e., not significant at 95% but significant at 90%.

(2) Adding only HEAD+DEPREL to the dependency-free baseline produces a small, non-significant gain of +0.12 macro-F1 (CI [-1.08, 1.32], p=0.4480).

(3) Crucially, adding the derived dependency features (head POS, relative position, depth, subtree size, root indicator) on top of HEAD+DEPREL yields a statistically significant improvement of ,+0.75 macro-F1; the CI excludes 0 ,[0.01, 1.68] with p=0.0237.

Interpretation: Overall, the evidence indicates that syntactic information helps, but the useful signal is distributed across multiple dependency-derived features rather than being captured by HEAD and DEPREL alone. The non-significant result in (1) at the 95% level likely reflects the variance of macro-F1, which gives equal weight to rare tags; nevertheless, the gain is positive and borderline (significant at 90%), suggesting a consistent upward trend. Result (2) shows that HEAD+DEPREL alone are insufficient to move macro-F1 reliably.

Result (3) demonstrates that derived structural features provide complementary information beyond HEAD+DEPREL, delivering a statistically reliable improvement. Taken together, these outcomes are not a negative result: they refine the conclusion to *which* dependency signals matter—namely, the combination of head/label with structural/positional context—rather than whether any single dependency cue suffices. This pattern is consistent with the intuition that POS disambiguation in code-mixed text benefits from both local syntactic anchors (HEAD/DEPREL) and their structural context (depth, subtree size, relative position, root cues).

For POS tagging on code-mixed Telugu–English, we recommend retaining the full dependency feature set. Future work with larger test sets may render comparison (1) significant at 95%.

6.4 Dependency Relation Weights

Table 15 shows the composite weights for each dependency relation, calculated from graph centrality, frequency, and LAS sensitivity.

Relations like OBJ, NSUBJ, and PARATAXIS have the highest weights because they are central in parse trees, occur moderately to frequently in code-mixed text, and strongly affect parsing accuracy when misclassified. For example, OBJ has the maximum normalized weight due to its key syntactic role and high error sensitivity. Relations like CASE, MARK, and PUNCT get moderate weights, reflecting their frequency but lower structural impact. Low-frequency labels like EXPL, REPARANDUM, and ORPHAN are down-weighted despite high sensitivity, as they have weak graph connectivity. This weighting strategy helps the model focus on important syntactic relations while giving less emphasis to unstable or rare ones.

Though we experimented with learnable relation weights, performance degraded due to overfitting in

the small dataset (4,152 sentences). Hence, manually computed weights based on empirical statistics are preferred for their stability and interpretability.

6.5 Robustness of Learned Head Bias

The language-aware head bias is tied to head–dependent assignment. It influences how the parser decides which token is the head of a given dependent. If the heads in the training data are imperfect, it would be directly perturbing the signal that teaches the bias weights. To examine the sensitivity of the learned head bias to annotation errors, we simulated annotation noise by randomly reassigning gold heads for 5%, 10%, and 15% of tokens in the training data. This allows us to see whether the bias can still learn stable preferences when the supervision signal for heads is imperfect. We also evaluated the sensitivity of the head–language bias mechanism to errors in predicted language tags by introducing controlled noise into the language tags. Noise is injected by randomly flipping the tag of a fixed percentage of tokens to one of the other possible values.

Table 16 compares the impact of controlled noise in head annotations versus language-tag annotations on parsing accuracy and the stability of the learned head–language bias. For head noise, both LAS and UAS decrease steadily with increasing noise, and Kendall's τ declines gradually from 1.0 to 0.783, indicating that although accuracy degrades, the ordering of bias weights remains fairly consistent under moderate noise (≤15%).

Language-tag noise shows a less monotonic trend in LAS/UAS degradation; small to moderate noise (5–15%) sometimes results in smaller LAS drops than head noise, but Kendall's τ falls faster (to 0.678 by 15%), reflecting greater sensitivity in bias weight ordering to incorrect language cues. At higher noise levels (20–25%), both accuracy and bias stability are more strongly affected, with Kendall's τ dropping to 0.51 in the worst case.

Overall, the learned head–language bias is more resilient to head-annotation errors than to large-scale language-tag errors, but parsing accuracy is impacted more severely by head noise. This suggests that while the bias mechanism can tolerate moderate language-ID noise, careful control of head-dependent annotations remains critical for parsing quality.

6.6 Impact of Code-Mixing Intensity on Parsing Accuracy

To assess whether our parser's improvements are consistent across different levels of code-mixing, we stratified the test set along CMI [42] value: code-mixing intensity, measured by CMI metric, with low-mixed sentences (CMI <= 0.35) and high-mixed sentences (CMI > 0.35). As shown in Table 17, performance is higher for sentences with lower or moderate code-mixing (CMI<=0.35), achieving 80.75 UAS and 67.49 LAS, compared to 75.89 UAS and 62.82 LAS for high-mixing cases (CMI>0.35). The drop in both UAS and LAS for high CMI suggests that heavy code-switching, with frequent shifts between Telugu and English, adds more structural and lexical variation, making head attachment and relation labeling harder. In lower/mid CMI sentences, the parser benefits from more stable syntactic patterns and clearer cues from contiguous monolingual segments, allowing the language-conditioned biases and dependency-type weights to operate more effectively. The relatively larger LAS drop in high-CMI sentences indicates that while heads can still be attached with reasonable accuracy, the fine-grained relation assignment suffers more in the presence of dense language alternation.

6.7 Effect of different embedding components and XLM-R layer-combination strategies on dependency parsing performance

While XLM-R's contextual embeddings implicitly encode morphosyntactic information, explicit POS tag embeddings offer a complementary bias that is especially beneficial for low-resource, noisy, and code-mixed text, where subword splits may be irregular and language switches produce syntactic patterns unseen during pretraining. Table 18 quantifies the contribution of each embedding type. Removing POS embeddings causes the largest drop in accuracy (LAS −11.45, UAS −7.14), confirming that explicit POS information is critical in our code-mixed parsing setting, likely due to its robustness against irregular subword segmentation and unseen language-switch patterns in the pretrained XLM-R model. Removing word embeddings or character embeddings leads to only small LAS drops (≈ −1.41 to −2.14), suggesting they provide modest but complementary lexical/morphological cues.

Removing all three non-contextual embeddings while retaining XLM-R causes a larger LAS drop (−5.72) than removing any single feature, highlighting their combined contribution. Conversely, removing XLM-R but keeping the three non-contextual embeddings yields only a small decrease in LAS (−0.82), indicating that these structured linguistic features can partially compensate when contextual embeddings are

absent. However, full performance still requires both sources of information.

We experimented with different configurations of XLM-R layers for generating contextualized embeddings, tuning the choice on the development set. Table 19 summarizes these results. While the bottom layers of transformer encoders are known to retain surface-level lexical and subword-level features, and mid-to-high layers capture syntactic information, the top layers in multilingual models such as XLM-R additionally encode strong cross-lingual alignment patterns. For code-mixed dependency parsing, this alignment is particularly useful in resolving head–dependent relations across language boundaries. Empirically, averaging the representations from the last four layers (layers -1,-2,-3,-4, where -1 denotes the final layer, -2 the second-to-last, and so on) yielded the highest UAS/LAS on the development set, outperforming configurations that used only layers (-2,-3,-4) which is Stanza's default, or all twelve layers.

6.8 Qualitative Examples

To illustrate the practical effect of head-language bias and dependency-derived features, Table 20 presents selected sentences from the test set where the full model (with head-language bias and relation-specific weights) outperformed the model without these enhancements. In each case, we show the gold label (which is also the prediction by the model with head-language bias and relation-specific weights) and the prediction from a baseline without these features, highlighting where the enhanced model led to the correct disambiguation.

Disambiguation of Subject--Object Confusion: Several cases (e.g., media, 5g) show that the no-bias model incorrectly assigns OBJ to words that should be NSUBJ. The enhanced model benefits from head-language bias, which leverages the head token's language identity to guide relation selection. For instance, in ``Social media Ila undentra...", recognizing that *undentra* (Telugu) is the governing verb and that *media* is a nominal subject is aided by cross-lingual syntactic cues.

Correct Identification of Sentence Heads: Tokens like *padthadi* and *cheyali* illustrate improvements in root detection. In both cases, the no-bias model either shifted the head to a nearby clause or incorrectly labeled a verb in a side-by-side, but different, sentence as the root, resulting in a PARATAXIS or ROOT swap.

Handling of Clause-Attachment Ambiguity: For *unde*, the no-bias model assigned ADVCL instead of ACL, indicating difficulty in determining whether the clause modifies a noun or a verb. The enhanced model, by combining head-language bias and relation-specific weighting, correctly attaches *unde* as an adjectival clause modifying *colour*, aligning with the intended structure.

Resolution of Modifier Role Confusion: Tokens like *my* and *Srh* reveal errors where modifiers (NMOD and NSUBJ) are misclassified. The enhanced model's access to syntactic patterns learned from relation-specific weights allows it to better differentiate between nominal dependents and arguments.

Improved Sequential Interpretation in Code-Mixed Contexts: Examples such as *Next* highlight discourse-level challenges. The no-bias model linked *Next* incorrectly to a preceding predicate (*akkuvindi*), whereas the enhanced model recognized its role as an adverbial modifier of the subsequent clause (*estaru*).

Overall Impact: The enhanced model better matches sentence structure to meaning, even with language mixing, informal grammar, and noisy text. Head-language bias helps choose the right relations between words in different languages, and relation-specific weights capture structural patterns unique to Telugu-English code-mixed text.

6.9 Cross-Domain Evaluation with Twitter Data

While the parser is trained and evaluated on YouTube comments, real-world code-mixed text occurs across diverse platforms. Twitter, in particular, presents greater structural noise due to its character limit, conversational markers, hashtags, and non-standard punctuation. Evaluating on a different domain helps measure the parser's robustness and generalization capacity, and highlights which dependency relations are most sensitive to domain shift.

To perform this evaluation, we curated a small corpus of approximately 300 Telugu-English code-mixed sentences from Twitter. Tweets were collected using relevant hashtags and keywords to ensure diversity in topics and styles. Each sentence was manually annotated following UD 2.0 guidelines, using the same process as for the YouTube dataset. This ensures comparability between the in-domain and out-of-domain results. Table 21 compares the in-domain (YouTube) and out-of-domain (Twitter) performance of the parser. On YouTube, the parser achieves UAS 76.12 / LAS 63.04, while on Twitter the scores decrease to UAS 71.48 / LAS 56.69. This corresponds to an absolute drop of 4.64 points in UAS and 6.35 points in LAS, showing that relation labeling is more sensitive to domain shift than head attachment. Table 22 shows

the relation-wise comparison. Frequent morpho-syntactic relations (e.g., AMOD, ADVCL, AUX) generalize well across domains, whereas structurally complex or low-frequency relations (e.g., CCOMP, CONJ, APPOS, CSUBJ, XCOMP) remain near zero across both domains, suggesting structural weaknesses in handling coordination and clausal embedding. Medium-frequency relations such as PARATAXIS, DISCOURSE, DET, VOCATIVE, PUNCT, and FLAT show measurable declines, highlighting the difficulty of handling Twitter's noisy punctuation, conversational style, and fragmented utterances. The ROOT relation, however, remains stable across domains (0.75 vs. 0.74 F1 score), showing that the parser reliably identifies sentence heads across domains.

Overall, the results suggest that while the parser generalizes reasonably well to frequent, morphologically transparent relations, it struggles with discourse phenomena, low-frequency structures, and domain-specific noise in Twitter data. These findings should not be interpreted as inherent limitations of the model itself, but rather as a reflection of the distinct characteristics of Twitter as a domain. To achieve strong performance on Twitter or similarly noisy domains, it is essential to either collect and annotate representative training data from those platforms or apply domain adaptation strategies such as fine-tuning, multi-domain training, or data augmentation. This would allow the model to better capture the unique linguistic and structural patterns of Twitter.

6.10 Robustness of POS Tagging to Noisy Dependency Parses

To assess robustness, we evaluated POS tagging with dependency features extracted from parses of varying quality, we calibrated noise injection parameters on the development set to produce three degradation levels in LAS: light (approx--10 LAS), medium (approx--20 LAS), and heavy (approx--30 LAS).

These settings are then applied to the predicted dependency parses of the test set before extracting dependency-based features for the POS tagger. As shown in Table 23, even under heavy corruption (LAS drop of --32.41), the POS F1 only decreased by ~ 1.77 points relative to the clean baseline.

This suggests that the syntactic signal is beneficial even in the presence of parsing errors, likely because head–dependent relations and dependency label distributions still encode useful positional and structural cues despite noise. These results reinforce that the observed POS tagging gains are not dependent on unrealistically high parse accuracy and are likely to transfer to settings where parsing quality is lower than in our experiments.

7. ERROR ANALYSIS

7.1 Dependency Parsing Confusion Patterns

To diagnose key sources of error, we analyze the confusion pairs between predicted and gold-standard dependency relations on the TEMPLE test dataset. Table 24 lists the ten most frequent confusion pairs along with example sentences.

The most frequent confusion, NSUBJ → OBJ (309 cases), shows ambiguity in core arguments, often due to free word order and missing case markers in Telugu-English code-mixed text. Reciprocal confusions (OBJ → NSUBJ) show that identifying arguments is challenging in both directions. Misclassifications such as NSUBJ → NMOD and FLAT → NMOD suggest the parser struggles to assign roles within noun phrases or named entities, especially without capitalization or case markers. Other misclassifications include ACL → ADVCL, showing difficulty separating noun-modifying from verb-modifying clauses, and CONJ → PARATAXIS, reflecting difficulty in detecting coordination from juxtaposed clauses in informal social media discourse. Misclassifying PARATAXIS as ROOT points to challenges in modeling clause structure in social media discourse, where main and independent clauses are casually mixed.

Since NSUBJ →OBJ is the most confused pair, we further analyzed this pair. To pinpoint the source of this confusion, we separated cases with the correct syntactic head but incorrect label from those with both head and label errors. We found that 89.97% of such confusions (278 out of 309) involved the correct head but an incorrect role label, indicating that the parser generally attaches arguments to the correct predicate but struggles with role assignment. Only 9.03% of cases reflected genuine attachment errors. POS patterns (Table 25) reveal a strong concentration in VERB–NOUN pairs (72.9%), with smaller contributions from VERB–PRONOUN (7.6%) and VERB–PROPER NOUN (6.3%). Further, in 66% (204 out of 309) instances, the POS tag of the immediate word following the NSUBJ is VERB (in the entire data, this figure is 30.23% i.e., 1,255 out of 4,152). This suggests that the model might be relying more on lexical cues or dependency label priors than on local syntactic patterns.

Overall, these confusing patterns indicate that the parser's limitation lies in:

- Low-frequency relations with sparse training data.
- Clause boundary and coordination detection in loosely punctuated, code-mixed text.
- Ambiguous head–dependent assignments within complex noun phrases.

These highlight the need for data augmentation of low-frequency relations, enhanced clause segmentation strategies, and head disambiguation features.

7.2 POS Tagging Confusion Patterns

The confusion matrix for POS tagging on the ICON test dataset is presented in Figure 3. We examine both the tags with the highest accuracy and the most frequently confused tag pairs.

Table 26 lists the POS tags with accuracy above 90%. These include punctuation (B-PUNCT), symbols (SYM), adpositions (B-ADP), and content categories like proper nouns (B-PROPN), common nouns (B-NOUN), pronouns (B-PRON), and verbs (B-VERB), suggesting robust handling of unambiguous tokens.

The most frequent POS tag confusions in Table 27 reflect the morphological ambiguity, and code-switching patterns characteristic of Telugu-English social media text.

- VERB → NOUN errors: These confusions arise when English verbs are used in their bare stem form (e.g., hit, pick, love, drive, post, tweet, reply), without tense or aspect markers. In code-mixed sentences, these stems often look identical to English nouns, so the tagger tends to predict NOUN.
- PROPN → NOUN errors: In many cases, proper nouns are not capitalized (e.g., yuvi), especially in Roman-script Telugu–English social media writing. Without uppercase cues, the model tends to classify them as NOUN. Also, words that are common nouns in English but used as named entities are often tagged as NOUN because the model defaults to their most frequent sense e.g., Mother (person name "Mother Teresa"), Police (movie title “Police Story”), House (in brand "Mansion House").
- NOUN → VERB errors: English nouns in -ing form (gerunds) are often misinterpreted as present participles, especially without determiners or prepositions to mark them as verbs (e.g., crying, writing). Also, nouns like 'release' (film launch event), 'comment' (a posted message), 'update' (new announcement) are ambiguous because they also serve as verbs in English.
- NOUN → PROPN errors: Single characters, abbreviations, or letter-based short forms are often mistaken for proper nouns because they resemble shorthand for person references in chat style (e.g., 'K' for thousand, 'rt' for retweet).
- ADJ → NOUN errors: Some adjectives are used in a noun-like role to refer to a person, group, or quality, a common feature in informal speech (e.g., '*adevado pilla* fan'. '*pilla*' means small girl and is normally a noun but here it is an adjective meaning 'small').
- ADV → ADJ errors: Adverbs that modify verbs or adjectives, especially degree words, are often tagged as adjectives when they lack explicit morphological marking (e.g., '*chaala istam*', '*ekkuva disturbing ga vundi*'). Also, adverbs expressing manner or emphasis may be mistaken for adjectives when used before nouns or as standalone emphatic words (e.g., '*gattiga ayindi*')
- ADV → PRON errors: Many Telugu adverbs of manner, extent, or emphasis, especially when written in Roman script, look similar to interrogative or indefinite pronouns, leading to confusion e.g., *edaina* (‘anyhow’), *emana* (‘somehow’). Also, time-related adverbs can be misread as pronominal references e.g., *inthe* (‘like this'), *ryt* (‘right now’).
- SCONJ → ADP errors: 13 out of 15 misclassifications of subordinating conjunctions as adpositions is due to the word 'ani'. The Telugu word 'ani' is a highly frequent complementizer introducing reported speech, thoughts, or perceptions. In Roman-script code-mixing, its fixed position after quoted or paraphrased content can resemble an adpositional function to a POS tagger.
- NOUN → ADJ errors: Nouns expressing abstract qualities, measurements, or statistical values can appear before other nouns as modifiers, leading the model to tag them as ADJ e.g., test average, meaning average in cricket test matches. Also, nouns used in idiomatic or cultural expressions often appear before another noun, mimicking adjective placement (e.g., 'time *waste*', '*dressing* sense').
- X → PROPN errors: All 13 cases of X ('Other' category POS tag) misclassified as proper nouns are non-Telugu words such as Tamil and Spanish (e.g., 'me sigo tentando nenu forra').

These findings suggest that performance could be improved by normalizing transliterated words, adding morphological features, and restoring capitalization to better identify proper names.

8. FUTURE WORK

Telugu has rich morphology, with many case endings for nouns and verb endings that show agreement, tense, aspect, and mood. This morphology can help the parser distinguish between relations NSUBJ vs. OBJ or OBL vs. NMOD, especially in code-mixed text where English words often lack clear case markers. While our current feature set includes POS tags, embeddings, and language tags, we did not incorporate explicit case or morphological tags in the main experiments to avoid dependency on external morphological analyzers, which are scarce for informal Telugu. High-quality morphological annotation could further benefit dependency parsing of Telugu–English code-mixed text, and is a promising direction for future work.

While our experiments focus on Telugu-English, the proposed methodology is designed to be generalizable to other code-mixed language pairs with appropriate adaptation. The key components — manual UD 2.0 annotation, integration of language tags into the parsing model, and dependency relation weighting — are language-agnostic in principle, provided that annotated data is available.

Although this study focuses on Telugu-English data, many syntactic properties (e.g., agglutination, postpositions, SOV order) are shared across Dravidian languages. This suggests potential transferability of the parsing model to Kannada-English, Tamil-English, or Malayalam-English data. However, the lack of POS-tagged datasets for these language pairs currently limits empirical evaluation. Publicly available data is sparse or lacks syntactic annotations. For instance, although [22] reports the creation of a Kannada-English POS-tagged corpus, the dataset is not publicly available. The COLI-Kanglish dataset [43] provides only language identification tags without POS annotations. The data from [44] provides language identification tags only for Malayalam-English code-mixed text. For Tamil-English and Malayalam-English, to our knowledge, there are no publicly available POS-tagged code-mixed corpora. Future work can explore zero-shot or few-shot adaptation using cross-lingual embeddings or joint training on multilingual code-mixed data.

Novel Contributions, Incremental Advances, and Best Practices: This study contributes both new knowledge and incremental methodological advancements to the field of code-mixed NLP. In terms of new knowledge, the creation of a manually annotated Telugu-English dependency treebank enables, for the first time, systematic analysis of cross-lingual syntactic structures in this language pair. The empirical findings reveal how grammatical relations behave across language boundaries and provide insights into challenges such as argument ambiguity, head-direction variability, and clause structuring in informal code-mixed text.

Additionally, the proposed language-aware head-dependent bias offers new evidence that language identity plays a critical role in syntactic head selection in code-mixed contexts. This extends prior assumptions in multilingual parsing by demonstrating that explicitly modeling language-conditioned preferences leads to measurable improvements. Similarly, the analysis of dependency relation-specific weighting highlights that structurally central relations contribute disproportionately to parsing accuracy, providing a new perspective on incorporating syntactic importance into model design.

At the same time, certain aspects of this work represent incremental advancements built upon established architectures. The use of a biaffine parser and BiLSTM-CRF model follows prior work; however, the contribution lies in adapting these frameworks to code-mixed settings through linguistically informed modifications and feature integration. These enhancements demonstrate that even within existing architectures, targeted adaptations can yield meaningful improvements when guided by domain-specific insights.

The findings of this study also suggest several best practices for future research in code-mixed NLP. First, the availability of high-quality, manually annotated datasets remains essential, even in the era of large language models. Second, incorporating language-aware mechanisms is crucial for capturing cross-lingual syntactic patterns. Third, combining core dependency features with structural and positional information is more effective than relying on isolated features. Finally, integrating syntactic information into downstream tasks can improve performance, particularly in structurally complex and low-resource settings.

Overall, this work demonstrates that advancing code-mixed NLP requires both the creation of reliable linguistic resources and the development of models that explicitly account for cross-lingual structure. By combining these elements, the study provides a foundation for future research while also offering practical guidance for building robust multilingual systems.

9. CONCLUSION

This work addresses a critical gap in code-mixed NLP, namely the lack of syntactic resources and specialized models for Telugu-English code-mixed text, which has limited progress in reliable syntactic analysis for this language pair. To overcome this limitation, we introduced a manually annotated dependency treebank and proposed targeted enhancements to a biaffine dependency parser that explicitly model cross-lingual syntactic interactions.

The experimental results demonstrate that incorporating language-aware head-dependent bias and relation-specific structural weighting leads to consistent improvements in both head attachment and relation labeling accuracy. These findings confirm that explicitly modeling language identity and structural importance is essential for handling the variability inherent in code-mixed text, where conventional monolingual or multilingual parsers fall short.

Furthermore, the integration of dependency-derived syntactic features into POS tagging provides empirical evidence that syntactic structure contributes meaningfully to sequence labeling tasks in code-mixed settings. Importantly, the results show that the benefit is not derived from dependency labels alone, but from their combination with structural features such as relative position and tree-based context, highlighting the role of global syntactic information in resolving ambiguity.

Taken together, these findings demonstrate that addressing resource scarcity and incorporating linguistically informed modeling are both necessary for improving syntactic analysis of code-mixed text. The proposed treebank enables reliable training and evaluation, while the model adaptations provide a principled approach to capturing cross-lingual dependencies.

Beyond the immediate improvements in parsing and tagging performance, this study contributes to knowledge creation by providing insights into how syntactic structure operates in Telugu-English code-mixed text and by establishing a foundation for future research in low-resource multilingual NLP. The results suggest that similar language-aware and structurally guided approaches can be extended to other code-mixed language pairs, thereby supporting the development of more robust and generalizable multilingual NLP systems.

Open Research Issues: While this study advances syntactic analysis for Telugu-English code-mixed text, several open research issues remain. First, the scalability of manually annotated dependency treebanks is a significant challenge. Although the proposed dataset provides a strong foundation, extending annotation to larger and more diverse corpora is necessary to capture the full variability of real-world code-mixed language.

Second, the current approach is evaluated primarily on YouTube and a limited Twitter dataset. The observed performance drop across domains highlights the need for robust domain adaptation techniques and multi-domain training strategies to improve generalization in diverse social media contexts.

Third, the handling of high code-mixing intensity remains an open challenge. As shown in the analysis, parsing accuracy decreases with increasing language switching, indicating that current models still struggle with dense cross-lingual interactions. Developing models that can dynamically adapt to varying levels of code-mixing is an important direction for future work.

Finally, recent advances in large language models suggest opportunities for semi-automatic annotation and cross-lingual transfer. However, effectively combining LLM-based methods with high-quality manual annotations remains an open research question, particularly in ensuring linguistic consistency and reducing annotation noise.

Addressing these open issues is essential for advancing robust, scalable, and generalizable syntactic modeling of code-mixed text and for supporting the development of next-generation multilingual NLP systems.

AUTHOR CONTRIBUTIONS:

SM and VRS conceived and designed research. SM wrote the main manuscript, performed the experiments and created the figures for the manuscript. VRS is the supervisor for the PhD project of SM. He reviewed the manuscript and advised on the experiments. Both authors read and approved the manuscript.

REFERENCES:

[1] Carol Myers-Scotton. 1993. Common and uncommon ground: Social and structural factors in codeswitching. *Language in society* 22, 4 (1993), 475–503.

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

[3] Marie-Catherine De Marneffe, Christopher D Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal dependencies. *Computational linguistics 47, 2 (2021), 255–308.*

[4] Irshad Bhat, Riyaz Ahmad Bhat, Manish Shrivastava, and Dipti Misra Sharma. 2018. Universal Dependency Parsing for Hindi-English Code- Switching. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. 987–998.

[5] Urmi Ghosh, Dipti Misra Sharma, and Simran Khanuja. 2019. Dependency parser for bengali-english code-mixed data enhanced with a synthetic treebank. In *Proceedings of the 18th International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2019)*. 91–99.

[6] Slav Petrov, Dipanjan Das, and Ryan McDonald. 2011. A universal part-of-speech tagset. *arXiv preprint arXiv:1104.2086* (2011).

[7] Joakim Nivre. 2006. *Inductive dependency parsing*. Springer.

[8] Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajic. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of human language technology conference and conference on empirical methods in natural language processing*. 523–530.

[9] Danqi Chen and Christopher D Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 740–750.

[10] Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Trans- actions of the Association for Computational Linguistics* 4 (2016), 313–327.

[11] Timothy Dozat and Christopher D Manning. 2017. Deep Biaffine Attention for Neural Dependency Parsing. In *International Conference on Learning Representations*.

[12] Dan Kondratyuk and Milan Straka. 2019. 75 languages, 1 model: Parsing universal dependencies universally. *arXiv preprint arXiv:1904.02099 (2019).*

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[14] Kellert, O., Tyagi, N., Imran, M., Licona-Guevara, N., & Gómez-Rodríguez, C. (2025). Parsing the Switch: LLM-Based UD Annotation for Complex Code-Switched and Low-Resource Languages. *arXiv preprint arXiv:2506.07274*.

[15] Mohamed, A., Zhang, Y., Vazirgiannis, M., & Shang, G. (2025). Lost in the mix: Evaluating llm understanding of code-switched text. *arXiv preprint arXiv:2506.14012*.

[16] Yuan Zhang and David Weiss. 2016. Stack-propagation: Improved Representation Learning for Syntax. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

[17] Şaziye Betül Özateş, Arzucan Özgür, Tunga Güngör, and Özlem Çetinoğlu. 2022. Improving Code-Switching Dependency Parsing with Semi- Supervised Auxiliary Tasks. In *Findings of the Association for Computational Linguistics: NAACL 2022*. 1159–1171.

[18] Akshith Putta. 2024. A dependency parser for code-switched Telugu-English. *Journal of High School Science* 8, 2 (2024), 143–155.

[19] S Nagesh Bhattu, Satya Krishna Nunna, Durvasula VLN Somayajulu, and Binay Pradhan. 2020. Improving code-mixed POS tagging using code- mixed embeddings. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)* 19, 4 (2020), 1–31.

[20] Yogarshi Vyas, Spandana Gella, Jatin Sharma, Kalika Bali, and Monojit Choudhury. 2014. Pos tagging of english-hindi code-mixed social media content. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 974–979.

[21] Tathagata Raha, Sainik Kumar Mahata, Dipankar Das, and Sivaji Bandyopadhyay. 2020. Development of pos tagger for english-bengali code-mixed data. *arXiv preprint arXiv:2007.14576* (2020).

[22] Abhinav Reddy Appidi, Vamshi Krishna Srirangam, Darsi Suhas, and Manish Shrivastava. 2020. Creation of corpus and analysis in code-mixed Kannada-English social media data for POS tagging. In *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*. 101–107.

[23] Anupam Jamatia, Björn Gambäck, and Amitava Das. 2016. Collecting and annotating indian social media code-mixed corpora. In *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, 406–417.

[24] Prakash B Pimpale and Raj Nath Patel. 2016. Experiments with POS tagging code-mixed Indian social media text. *arXiv preprint arXiv:1610.09799 (2016).*

[25] Rupal Bhargava, Bapiraju Vamsi Tadikonda, and Yashvardhan Sharma. 2016. BITS_Pilani_Team1@ POS Tagging for Code Mixed Indian Social Media. In *International Conference on Natural Language Processing*.

[26] Raj Nath Patel, Prakash B Pimpale, and M Sasikumar. 2016. Recurrent neural network based part-of-speech tagger for code-mixed social media text. *arXiv preprint arXiv:1611.04989* (2016).

[27] Ayan Sengupta, Sourabh Kumar Bhattacharjee, Tanmoy Chakraborty, and Md Shad Akhtar. 2021. HIT: A hierarchically fused deep attention network for robust code-mixed language representation. *arXiv preprint arXiv:2105.14600* (2021).

[28] Suman Dowlagar and Radhika Mamidi. 2021. A pre-trained transformer and CNN model with joint language ID and part-of-speech tagging for code-mixed social-media text. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*. 367–374.

[29] Deepak Gupta, Shubham Tripathi, Asif Ekbal, and Pushpak Bhattacharyya. 2017. SMPOST: parts of speech tagger for code-mixed indic social media text. *arXiv preprint arXiv:1702.00167* (2017).

[30] Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. 2016. A joint many-task model: Growing a neural network for multiple NLP tasks. *arXiv preprint arXiv:1611.01587* (2016).

[31] Houquan Zhou, Yu Zhang, Zhenghua Li, and Min Zhang. 2020. Is POS tagging necessary or even helpful for neural dependency parsing?. In *CCF International Conference on Natural Language Processing and Chinese Computing. Springer, 179–191.*

[32] Sandeep Maddu and Vizianandа Row Sanapala. 2025. Corpus Creation of Telugu-English Code-Mixed Text Using BiLSTM-CRF Model with Phonetic Encoding, and Design of Linguistic Turbulence Index as a New Code-Mixing Metric. doi:10.5281/zenodo.16540545

[33] Peng Qi, Timothy Dozat, Yuhao Zhang, and Christopher D Manning. 2019. Universal dependency parsing from scratch. *arXiv preprint arXiv:1901.10457* (2019).

[34] Timothy Dozat, Peng Qi, and Christopher D Manning. 2017. Stanford's graph-based neural dependency parser at the CoNLL 2017 shared task. In *Proceedings of the CoNLL 2017 shared task: Multilingual parsing from raw text to universal dependencies*. 20–30.

[35] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. *arXiv preprint arXiv:2003.07082* (2020).

[36] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116* (2019).

[37] Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991* (2015).

[38] Jie Yang and Yue Zhang. 2018. NCRF++: An Open-source Neural Sequence Labeling Toolkit. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. http://aclweb.org/anthology/P18-4013

[39] Yash Madhani, Sushane Parthan, Priyanka Bedekar, Gokul Nc, Ruchi Khapra, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh M Khapra. 2023. Aksharantar: Open Indic-language transliteration datasets and models for the next billion users. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. 40–57.

[40] Margaret K.O. Russell R.C. 1918, 1922. US Patent 1262167, 1435663.

[41] Bradley Efron and Robert J Tibshirani. 1994. *An introduction to the bootstrap*. Chapman and Hall/CRC.

[42] Björn Gambäck and Amitava Das. 2014. On measuring the complexity of code-mixing. In *Proceedings of the 11th international conference on natural language processing, Goa, India*. 1–7.

[43] Hosahalli Lakshmaiah Shashirekha, Fazlourrahman Balouchzahi, Mudoor Devadas Anusha, and Grigori Sidorov. 2022. Coli-machine learning approaches for code-mixed language identification at the word level in kannada-english texts. *arXiv preprint arXiv:2211.09847* (2022).

[44] S Thara and Prabaharan Poornachandran. 2021. Transformer based language identification for malayalam-english code-mixed text. *IEEE Access* 9 (2021), 118837–118850.

[45] [24] S. Maddu, "TEMPLE corpus," Aug. 2025, accessed: Dec. 31, 2025. [Online]. Available: https://github.com/sandeep-maddu/templecorpus

[46] Anouck Braggaar and Rob Van Der Goot. 2021. Challenges in annotating and parsing spoken, code-switched, Frisian-Dutch data. In *Proceedings of the Second Workshop on Domain Adaptation for NLP*. 50–58.

[47] Max Müller-Eberstein, Rob Van Der Goot, and Barbara Plank. 2021. Genre as weak supervision for cross-lingual dependency parsing. *arXiv preprint arXiv:2109.04733* (2021).

Table 1: A sample UD-style CoNLL-U annotation for the Telugu-English code-mixed sentence "Anna N5 food plaza loo bill challa akkuvaa and quantity thakuvu anna". Each row corresponds to a token. The HEAD column indicates the syntactic head of the token, represented as the ID of the head word (with 0 reserved for the root). The DEPREL column specifies the dependency relation between the token and its head (e.g., NSUBJ for nominal subject, VOCATIVE for vocative modifier, FIXED for multiword expressions). For instance, in row~2, the token ``N5'' has HEAD = 6 and DEPREL = NMOD, which means that ``N5'' functions as a nominal modifier of the word ``bill'' (token~6). The MISC column encodes language tags: 0 = Telugu, 1 = English, 2 = Named Entity, 3 = Other.

ID	FORM	LEMMA	UPOS	XPOS	FEATS	HEAD	DEPREL	DEPS	MISC
1	Anna	_	NOUN	_	_	2	VOCATIVE	_	0
2	N5	_	PROPN	_	_	6	NMOD	_	2
3	food	_	NOUN	_	_	2	FIXED	_	1
4	plaza	_	NOUN	_	_	3	FIXED	_	1
5	loo	_	ADP	_	_	2	CASE	_	0
6	bill	_	NOUN	_	_	8	NSUBJ	_	1
7	challa	_	ADV	_	_	8	ADVMOD	_	0
8	akkuvaa	_	ADJ	_	_	0	ROOT	_	0
9	and	_	CCONJ	_	_	11	CC	_	1
10	quantity	_	NOUN	_	_	11	NSUBJ	_	1
11	thakuvu	_	ADJ	_	_	8	CONJ	_	0
12	anna	_	NOUN	_	_	11	VOCATIVE	_	0

Table 2: Train/dev/test split of the annotated Telugu-English code-mixed dataset.

Split	Sentences	Tokens
Train	2,903	36,184
Dev	412	5,180
Test	837	10,305
Total	4,152	51,669

Table 3: Counts and proportions of dependency relation tags in the TEMPLE dataset

Dependency relation	Description	Count	Percentage
CASE	case marking element	4,456	8.62
NSUBJ	nominal subject	4,398	8.51
ROOT	root of the sentence	4,203	8.13
OBJ	object of the verb	4093	7.92
PARATAXIS	side-by-side clause	3501	6.78
NMOD	nominal modifier	3351	6.49
VOCATIVE	addressed person	3202	6.20
OBL	oblique nominal	3109	6.02
ADVMOD	adverbial modifier	2958	5.72
PUNCT	punctuation	2613	5.06
AMOD	adjectival modifier	2468	4.78
DISCOURSE	interjection, conversation marker	1829	3.54
AUX	auxiliary	1635	3.16
ADVCL	adverbial clause modifier	1415	2.74
DET	determiner	1136	2.20
COMPOUND	compound word	1038	2.01
FLAT	fixed/name-like expression	840	1.63
CONJ	conjunct	683	1.32
NUMMOD	numeric modifier	663	1.28
XCOMP	open clausal complement	643	1.24
ACL	clausal modifier of noun	545	1.05
CC	coordinating conjunction	542	1.05
MARK	subordinating marker	445	0.86
CCOMP	clausal complement	353	0.68

LIST	chains of comparable items	331	0.64
GOESWITH	part of split word	317	0.61
FIXED	fixed multiword expression	230	0.45
APPOS	appositional modifier	230	0.45
IOBJ	indirect object	180	0.35
CSUBJ	clausal subject	118	0.23
COP	copula	113	0.22
DEP	unspecified dependency	21	0.04
EXPL	expletive	7	0.01
REPARANDUM	overridden disfluency	2	0.00
ORPHAN	remnant in ellipsis	1	0.00
DISLOCATED	displaced element	-	0.00
CLF	classifier	-	0.00
Total		51,669	100.00

Table 4: Counts and proportions of Universal POS tags in the TEMPLE dataset

POS tag	Description	Count	Percentage
NOUN	noun	14,526	28.11
VERB	verb	10,462	20.25
ADP	adposition	4,023	7.79
PROPN	proper noun	3,796	7.35
ADJ	adjective	3,557	6.88
PRON	pronoun	3,387	6.56
ADV	adverb	2,901	5.61
PUNCT	punctuation	2,478	4.80
INTJ	interjection	1,679	3.25
DET	determiner	1,367	2.65
NUM	numeral	976	1.89
PART	particle	737	1.43
CCONJ	coordinating conjunction	547	1.06
SCONJ	subordinating conjunction	414	0.80
X	other	450	0.87
AUX	auxiliary	291	0.56
SYM	symbol	78	0.15
Total		51,669	100.00

Table 5: Train/dev/test split of the re-annotated ICON 2016 POS-tagged dataset.

Split	Sentences	Tokens
Train	1,387	20,629
Dev	216	2,916
Test	377	5,926
Total	1,980	29,471

Table 6: POS tag-wise token count and percentage in the re-annotated ICON 2016 dataset, following Universal Dependencies 2.0 guidelines

POS tag	Count	Percentage
NOUN	6,201	21.04
VERB	4,403	14.94
PROPN	3,672	12.46
PUNCT	2,553	8.66
PRON	2,343	7.95
ADV	2,217	7.52

ADP	2,145	7.28
ADJ	2,095	7.11
SYM	1,161	3.94
INTJ	514	1.74
NUM	463	1.57
X	411	1.39
CCONJ	406	1.38
PART	283	0.96
DET	265	0.90
AUX	197	0.67
SCONJ	142	0.48
Total	29,471	100.00

Table 7: Hyperparameters used for training.

Hyperparameter	Value
BiLSTM hidden state dimension	300 per direction (3 layers)
Word/Character embedding size	100
POS tag embedding size	50
Contextual embedding	XLM-RoBERTa (top 4 layers)
Optimizer	Adam
Learning rate	0.002
Dropout	0.33

Table 8: Comparison of LAS and UAS across code-mixed dependency parsers for different language pairs.

Language Pair	Paper	No: of sentences	Approach	UAS (%)	LAS (%)
Telugu-English	Our work	4,152	Biaffine parser with relation weights and head-language bias	75.53	61.86
Telugu-English	[18]	300	UDify (multilingual BERT-based parser)	30.61	11.79
Hindi-English	[4]	1,898	Stack-based parsing with normalization and POS tags	80.23	71.03
Bengali-English	[5]	500	Stack based parsing with synthetic treebank and monolingual transfer	76.24	61.41
Turkish-German	[17]	2,184	Semi-supervised sequence labeling	70.92	61.65
Hindi-English	[17]	1,898	Semi-supervised sequence labeling	82.75	74.09
Komi-Russian	[17]	214	Semi-supervised sequence labeling	65.7	47.13
Frisian-Dutch	[17]	400	Semi-supervised sequence labeling	74.69	56.39
Frisian-Dutch	[46]	400	Deep biaffine parser (MaChAmp)	70.2	55.60
Hindi-English	[47]	1,800	Genre-based data selection	73.62	62.66
Turkish-German	[47]	1,891	Genre-based data selection	66.75	55.04

Table 9: Dependency parsing performance under ablation settings.

Setting	UAS	LAS	UAS Δ	LAS Δ
Full model (Base)	76.12	63.04	-	-
No relation weights	74.36	61.04	-1.76	-2.00
No head-language bias	75.31	61.74	-0.81	-1.30
No weights + no bias	74.01	61.31	-2.11	-1.73

Table 10: POS tagging performance across datasets.

Dataset	Accuracy	Precision	Recall	F1 Score
ICON (UD tagset)	91.76	86.01	83.37	83.73
TEMPLE (UD tagset)	86.24	80.59	79.18	78.16
ICON (Universal tagset)	69.93	76.86	65.4	67.51

Table 11: F1 score drop in POS tagging due to feature group removal.

Feature Dropped	F1 Score	F1 Drop
None (Full)	84.02	-
Lexical	82.55	−1.47
Head + Deprel	82.64	−1.38
Head + HeadPOS	82.92	−1.10
Language ID	83.03	−0.99
Structural tree features	83.05	−0.97
Head	83.10	−0.92
Syntactic	83.14	−0.88
Deprel	83.90	−0.12
Contextual	83.97	−0.05

Table 12: Statistical significance testing of UAS and LAS improvements using paired bootstrap resampling (B = 10,000). Scores are in absolute percentage points. Δ denotes the absolute difference between the enhanced and baseline systems. p_{one} is the *one-sided* p-value for the hypothesis that the enhanced system outperforms the baseline. p_{two} is the *two-sided* p-value for the hypothesis that there is any difference between systems (bever or worse)..

Metric	Δ	95% CI	p_{one}	p_{two}
UAS	+2.11	[+1.36 +2.86]	<0.001	0.0002
LAS	+1.73	[+1.00 +2.45]	<0.001	0.0002

Table 13: Per-relation LAS scores for the full model LAS_Bias (with head-language bias and dependency-relation-specific weights) versus the variant without bias and weights LAS_NoBias. *Delta* shows the absolute LAS difference (positive = improvement with bias), *p_McNemar* reports the McNemar's test *p*-value, and *Delta_CI* is the 95% confidence interval via bootstrapping

Relation	Count	LAS_Bias	LAS_NoBias	Delta	p_McNemar	Delta_CI
NMOD	744	0.66	0.44	0.23	0	[+0.19, +0.26]
NUMMOD	159	0.85	0.76	0.09	0.0066	[+0.03, +0.14]
MARK	86	0.59	0.51	0.08	0.2478	[-0.04, +0.20]
DET	220	0.79	0.72	0.07	0.0259	[+0.01, +0.14]
GOESWITH	69	0.07	0.01	0.06	0.2188	[-0.01, +0.13]
NSUBJ	951	0.36	0.3	0.05	0.0001	[+0.03, +0.08]
AMOD	413	0.86	0.81	0.05	0.0005	[+0.02, +0.08]
ADVCL	254	0.53	0.49	0.04	0.2203	[-0.02, +0.10]
ROOT	854	0.74	0.72	0.02	0.0662	[+0.00, +0.04]
DISCOURSE	369	0.6	0.59	0.01	0.5682	[-0.02, +0.05]
XCOMP	119	0.01	0	0.01	1.0000	[+0.00, +0.03]
CASE	873	0.91	0.91	0.01	0.3833	[-0.01, +0.02]
ADVMOD	630	0.71	0.71	0.00	0.7914	[-0.02, +0.03]

VOCATIVE	692	0.69	0.69	0.00	0.8424	[-0.02, +0.03]
IOBJ	29	0.00	0.00	0.00	1.0000	[+0.00, +0.00]
APPOS	32	0.00	0.00	0.00	1.0000	[+0.00, +0.00]
CCOMP	85	0.00	0.00	0.00	1.0000	[+0.00, +0.00]
COP	29	0.00	0.00	0.00	1.0000	[+0.00, +0.00]
LIST	42	0.00	0.00	0.00	1.0000	[+0.00, +0.00]
ORPHAN	1	0.00	0.00	0.00	1.0000	[+0.00, +0.00]
FIXED	44	0.00	0.00	0.00	1.0000	[+0.00, +0.00]
CSUBJ	23	0.00	0.00	0.00	1.0000	[+0.00, +0.00]
CONJ	148	0.00	0.00	0.00	1.0000	[+0.00, +0.00]
DEP	3	0.00	0.00	0.00	1.0000	[+0.00, +0.00]
PUNCT	517	0.94	0.95	-0.01	0.2962	[-0.04, +0.01]
PARATAXIS	689	0.62	0.63	-0.02	0.3149	[-0.04, +0.01]
COMPOUND	136	0.01	0.03	-0.02	0.25	[-0.05, +0.00]
OBL	657	0.64	0.67	-0.03	0.0226	[-0.05, -0.01]
ACL	97	0.03	0.06	-0.03	0.4531	[-0.09, +0.02]
OBJ	766	0.72	0.75	-0.03	0.0112	[-0.06, -0.01]
AUX	310	0.83	0.87	-0.05	0.0026	[-0.07, -0.02]
CC	111	0.45	0.52	-0.07	0.3663	[-0.22, +0.06]
FLAT	153	0.22	0.59	-0.37	0.0000	[-0.47, -0.28]

Table 14: Paired bootstrap results for POS tagging (macro-F1). Δ is (enhanced − baseline). CIs are 95% percentile intervals over B=10,000 sentence-level bootstrap replicates

Comparison	Metric	Baseline	Enhanced	Δ	95% CI	One-sided p
(1) None → All dep feats	Macro-F1	83.14	84.02	0.88	[-0.40, 2.20]	0.0969
(2) None → Only HEAD+DEPREL	Macro-F1	83.14	83.27	0.12	[-1.08, 1.32]	0.4480
(3) Only HEAD+DEPREL → All dep feats	Macro-F1	83.27	84.02	0.75	[0.01, 1.68]	0.0237

Table 15: Relation weights and their components used in the weighted dependency parsing model

Relation	Centrality	Frequency	Sensitivity	Relation weight
OBJ	1.62	0.08	0.19	1.00
ADVMOD	1.5	0.06	0.12	0.96
OBL	1.53	0.06	0.10	0.95
CONJ	1.53	0.01	0.10	0.93
NSUBJ	1.53	0.09	0.05	0.93
PARATAXIS	1.53	0.07	0.09	0.92
VOCATIVE	1.41	0.06	0.11	0.92
NMOD	1.5	0.06	0.09	0.92
ADVCL	1.53	0.03	0.11	0.92
CASE	1.35	0.09	0.14	0.89
DISCOURSE	1.38	0.04	0.06	0.88
ACL	1.32	0.01	0.11	0.80
GOESWITH	1.15	0.01	0.30	0.77
DET	1.18	0.02	0.07	0.77
XCOMP	1.26	0.01	0.21	0.75
PUNCT	1.06	0.05	0.14	0.75
AUX	1.24	0.03	0.09	0.74
ROOT	1.26	0.08	0.10	0.70
APPOS	1.12	0.00	0.19	0.68
MARK	1.03	0.01	0.13	0.63
CCOMP	1.15	0.01	0.27	0.63

AMOD	1.32	0.05	0.10	0.84
COMPOUND	1.32	0.02	0.12	0.84
LIST	1.29	0.01	0.08	0.78
NUMMOD	0.85	0.01	0.08	0.54
CSUBJ	0.94	0.00	0.14	0.53
FIXED	0.85	0.00	0.15	0.55
CC	0.79	0.01	0.00	0.52
IOBJ	0.71	0.00	1.00	0.46
EXPL	0.21	0.00	8.93	0.46
FLAT	0.74	0.02	0.05	0.47
COP	0.53	0.00	0.69	0.36
DEP	0.32	0.00	2.22	0.38
REPARANDUM	0.09	0.00	1.87	0.23
ORPHAN	0.09	0.00	0.00	0.43

Table 16: Robustness of the learned head–language bias to noisy head annotations and noisy language tags.

Type	Noise (%)	UAS	Δ UAS	LAS	Δ LAS	Avg |Δ bias|	Kendall τ
Head Noise	0	76.12	0.00	63.04	0.00	0.000	1.000
Head Noise	5	72.79	-3.33	59.74	-3.3	0.068	0.900
Head Noise	10	72.23	-3.89	58.47	-4.57	0.071	0.817
Head Noise	15	70.90	-5.22	57.4	-5.64	0.086	0.883
Head Noise	20	68.40	-7.72	55.04	-8.00	0.120	0.800
Head Noise	25	67.09	-9.03	53.72	-9.32	0.127	0.783
Lang Tag Noise	0	76.12	0.00	63.04	0.00	0.000	1.000
Lang Tag Noise	5	75.24	-0.88	59.87	-3.17	0.150	0.795
Lang Tag Noise	10	74.42	-1.70	61.46	-1.58	0.134	0.795
Lang Tag Noise	15	75.07	-1.05	62.29	-0.75	0.180	0.678
Lang Tag Noise	20	72.62	-3.50	58.64	-4.40	0.208	0.510
Lang Tag Noise	25	74.19	-1.93	60.88	-2.16	0.202	0.762

Table 17: Dependency parsing performance across sentences grouped by code-mixing intensity (CMI).

Code-Mixing Intensity (CMI)	UAS	LAS
CMI <= 0.35	80.75	67.49
CMI > 0.35	75.89	62.82

Table 18: Ablation study showing the impact of removing different embedding components.

Setting	UAS	LAS	Δ UAS	Δ LAS
Full model (XLM-R + POS + Word + Char)	76.12	63.04	0.00	0.00
POS removed	68.98	51.59	-7.14	-11.45
Word emb removed	74.44	61.63	-1.68	-1.41
Char emb removed	74.46	60.90	-1.66	-2.14
All three removed (only XLM-R)	74.10	57.32	-2.02	-5.72
XLM-R removed (POS + Word + Char only)	74.98	62.22	-1.14	-0.82

Table 19: Effect of different XLM-R layer-combination strategies on dependency parsing performance (UAS/LAS) for Telugu-English code-mixed parsing. Layer selection was tuned on the dev set.

Layer strategy	Dev Set		Test Set	
	UAS	LAS	UAS	LAS
Top 4 layers	73.47	60.79	76.12	63.04
Top layers (2,3,4) [Stanza default]	72.22	59.31	74.19	60.78
Top 1 layer	71.58	59.79	74.2	61.65
All 12 layers	72.24	59.05	74.17	60.50

Table 20: Qualitative examples where the full model corrects errors made by the no-bias model.

Word	Sentence	Head		Deprel	
		Gold	No Bias	Gold	No Bias
media	Social media Ila undentra Babu mugguri vedio tesi okkadi vedio edit chesaru	undentra	undentra	NSUBJ	OBJ
Next	KTR gadiki ahamkaram akkuvindi. Next Telangana people vote tho answer estaru	estaru	akkuvindi	ADVMOD	ADVMOD
padthadi	Anna world ride complete avvadaniki 2 years padthadi kada anna	ROOT	complete	ROOT	PARATAXIS
my	Congrats yesaswi daily okkasari aiyyana yesaswi song Vine vallu like kottandi my dear frds	frds	frds	NMOD	NSUBJ
5g	5g vastundhe bane undhe kani what about radiation and birds Anna	vastundhe	vastundhe	NSUBJ	OBJ
'Srh	Hyderabad lo main middle order kaavali bro..'Srh kachithanga poti patuthundhi	patuthundhi	poti	NSUBJ	NMOD
video	Anna camera focus correct ga ledu plz video correct ga pettu	pettu	pettu	OBJ	OBL
cheyali	Anna Nadi pocox2 mobile but na mobile camara dead ipoyendi dhaniki yemi cheyali anna	dead	ROOT	PARATAXIS	ROOT
unde	battery lo unde black colour enti danto light ela veluguthundi	colour	veluguthundi	ACL	ADVCL
Content	Content create chey bro already chesina gun videos ni malla malla endhuku	create	create	OBJ	NSUBJ

Table 21: Cross-domain evaluation results comparing YouTube (training domain) and Twitter (out-of-domain) datasets. Δ denotes the absolute difference between in-domain and out-of-domain performance.

Metric	YouTube (In-domain)	Twitter (Out-of-domain)	Δ
UAS	76.12	71.48	-4.64
LAS	63.04	56.69	-6.35

Table 22: Relation-wise dependency parsing performance (Precision, Recall, F1) on YouTube vs Twitter. ΔF1 is Twitter–YouTube.

Relation	P	R	F1	P	R	F1	ΔF1
ACL	0.43	0.03	0.06	0.25	0.05	0.08	+0.02
ADVCL	0.32	0.53	0.40	0.40	0.54	0.46	+0.06
ADVMOD	0.69	0.71	0.70	0.59	0.58	0.58	-0.12
AMOD	0.81	0.86	0.83	0.82	0.88	0.85	+0.02
APPOS	0.00	0.00	0.00	0.00	0.00	0.00	0.00
AUX	0.77	0.83	0.80	0.76	0.90	0.83	+0.03
CASE	0.88	0.91	0.90	0.83	0.90	0.87	-0.03
CC	0.42	0.45	0.43	0.40	0.33	0.36	-0.07
CCOMP	0.00	0.00	0.00	0.00	0.00	0.00	0.00
COMPOUND	0.50	0.01	0.01	0.00	0.00	0.00	-0.01
CONJ	0.00	0.00	0.00	0.00	0.00	0.00	0.00
COP	0.00	0.00	0.00	0.00	0.00	0.00	0.00
CSUBJ	0.00	0.00	0.00	0.00	0.00	0.00	0.00
DET	0.78	0.79	0.78	0.71	0.27	0.40	-0.38
DISCOURSE	0.60	0.60	0.60	0.45	0.52	0.48	-0.12
FLAT	0.43	0.22	0.29	0.00	0.00	0.00	-0.29
GOESWITH	0.38	0.07	0.12	0.00	0.00	0.00	-0.12
MARK	0.45	0.59	0.51	0.64	0.45	0.53	+0.02
NMOD	0.54	0.66	0.60	0.49	0.63	0.55	-0.05
NSUBJ	0.48	0.36	0.41	0.40	0.40	0.40	-0.01
NUMMOD	0.74	0.85	0.79	0.60	0.69	0.64	-0.15
OBJ	0.48	0.72	0.58	0.48	0.66	0.56	-0.02

OBL	0.50	0.64	0.56	0.44	0.60	0.51	-0.05
PARATAXIS	0.51	0.62	0.56	0.23	0.53	0.32	-0.24
PUNCT	0.93	0.94	0.93	0.94	0.58	0.72	-0.21
ROOT	0.76	0.74	0.75	0.74	0.74	0.74	-0.01
VOCATIVE	0.72	0.69	0.70	0.44	0.52	0.48	-0.22
XCOMP	1.00	0.01	0.01	0.00	0.00	0.00	-0.01
Overall	UAS = 76.12, LAS = 63.04			UAS = 71.48, LAS = 56.69			-6.35

Table 23: POS tagging performance on the test set under different noise levels in the predicted dependency parses. Noise levels were calibrated on the development set to produce approximately --10, --20, and --30 LAS drops.

Noise Level	LAS Drop	POS Acc}	POS F1	ΔF1 vs Base
Base (clean parses)	-	91.85	84.02	-
Light (25% head noise, 0% label noise)	-11.54	91.73	83.19	-0.83
Medium (0% head noise, 35% label noise)	-19.73	89.86	81.68	-2.34
Heavy (50% head noise, 30% label noise)	-32.41	90.20	82.25	-1.77

Table 24: Top 10 frequent relation confusions in the TEMPLE test dataset. The misclassified tokens are marked in red.

Confused Pair (Gold →Predicted)	Count	Example Sentence
NSUBJ → OBJ	309	Song baagundi. Okka lyric asamtrupti ga undi. full song vedio cheyyandi.
NSUBJ → NMOD	101	Arjun top five lo undali..game super ga aduthadu and Arjun smile super
FLAT → NMOD	96	Last time india china war lo india deniki odipoindo koda explain chesthe full clarity untundii
OBJ → OBL	85	Memu tayaru chestam maku help cheyyandi tel me phone number
PARATAXIS → ROOT	81	Suman TV variki request jaffer anchor ga failed plz Mee TRP padipothunnadi.
CONJ → PARATAXIS	74	Bro camara round ga vuntundhi but photos endhuku square ga vastay
NSUBJ →OBL	74	Bro aa song ne voice lo untai bagunu and next esp apudu vastadi
ACL →ADVCL	72	Teja sir please a curry ki a vessel vadalo oka video cheyandi pls
OBJ →NSUBJ	71	Bro money easy ga earn chestaru apps tho antaru kada nijamena bro explain please
NMOD →OBL	69	Annaa future and options paina incometax yala cheyali kastha chappandi Anna clarity ga pls

Table 25: Top POS patterns for NSUBJ} ↔ OBJ confusions (gold head POS, gold dependent POS).

POS Pair	Count	%
VERB-NOUN	277	72.9
VERB-PRON	29	7.6
VERB-PROPN	24	6.3
ADJ-NOUN	16	4.2
NOUN-NOUN	10	2.6

Table 26: POS tags with over 90% prediction accuracy on the ICON dataset.

POS Tag	Correct	Total	Accuracy (%)
PUNCT	458	467	98.07
SYM	194	202	96.04
ADP	387	405	95.56
PROPN	751	787	95.43
CONJ	54	57	94.74
NOUN	1,112	1179	94.32
NUM	76	81	93.83
PRON	479	513	93.37
VERB	863	930	92.80

Table 27: Top 10 most frequent POS tag confusions in ICON test dataset. Misclassified tokens are shown in red.

Confusion (Gold → Predicted)	Count	Example
B-VERB → B-NOUN	29	... Wd love to try new recipes.
B-PROPN → B-NOUN	21	... 2007 yuvi is back asalu sixes keka...
B-NOUN → B-VERB	19	... asalu last 15 mins superb writing.
B-NOUN → B-PROPN	18	... I know .. thankfully the biggest company I had worked for , had 7.5 K India lo
B-ADJ → B-NOUN	17	... eppudu start chesavu ... part time ah...
B-ADV → B-ADJ	15	... rate expansion mattiki gattiga ayindi...
B-ADV → B-PRON	15	... ryt now I am in second phase...
B-SCONJ → B-ADP	15	... ni feelings eppudu reverse lone untai ani maku telsu le nu kani inka
B-NOUN → B-ADJ	13	... odis nd tests avg okela untayaa naku telidu
B-X → B-PROPN	13	... Irukku (a) Naaku Inko Perundhi...

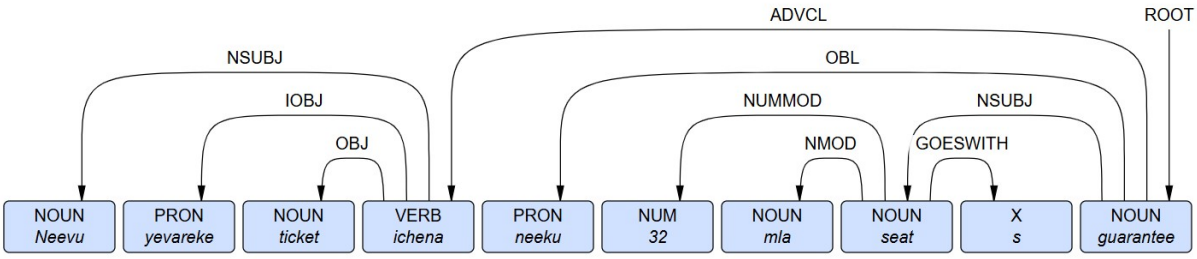

Figure 1: Dependency parse tree showing head-dependent relations for the sentence ``Neevu yevareke ticket ichena neeku 32 mla seats guarantee''. Arrows start at the head and point to the dependent. POS tags are shown below the words.

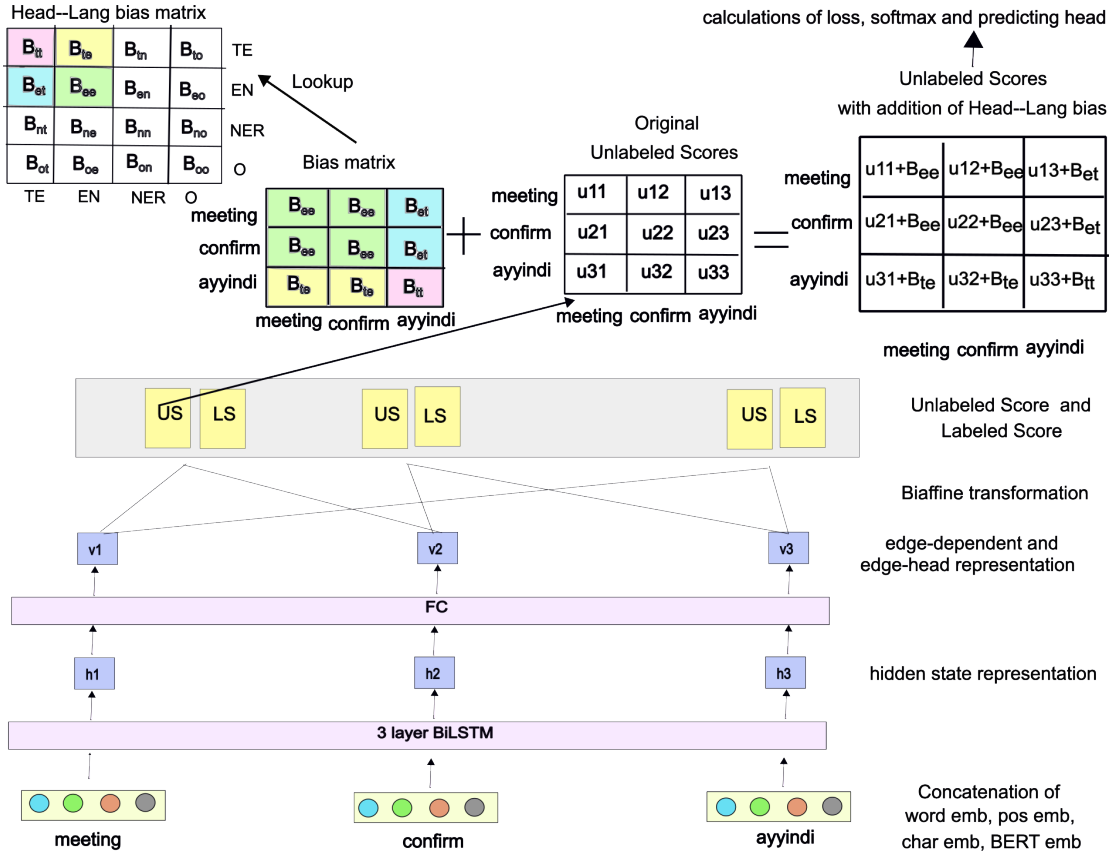

Figure 2: Addition of head lang bias to unlabeled scores in the deep biaffine parser of [33]

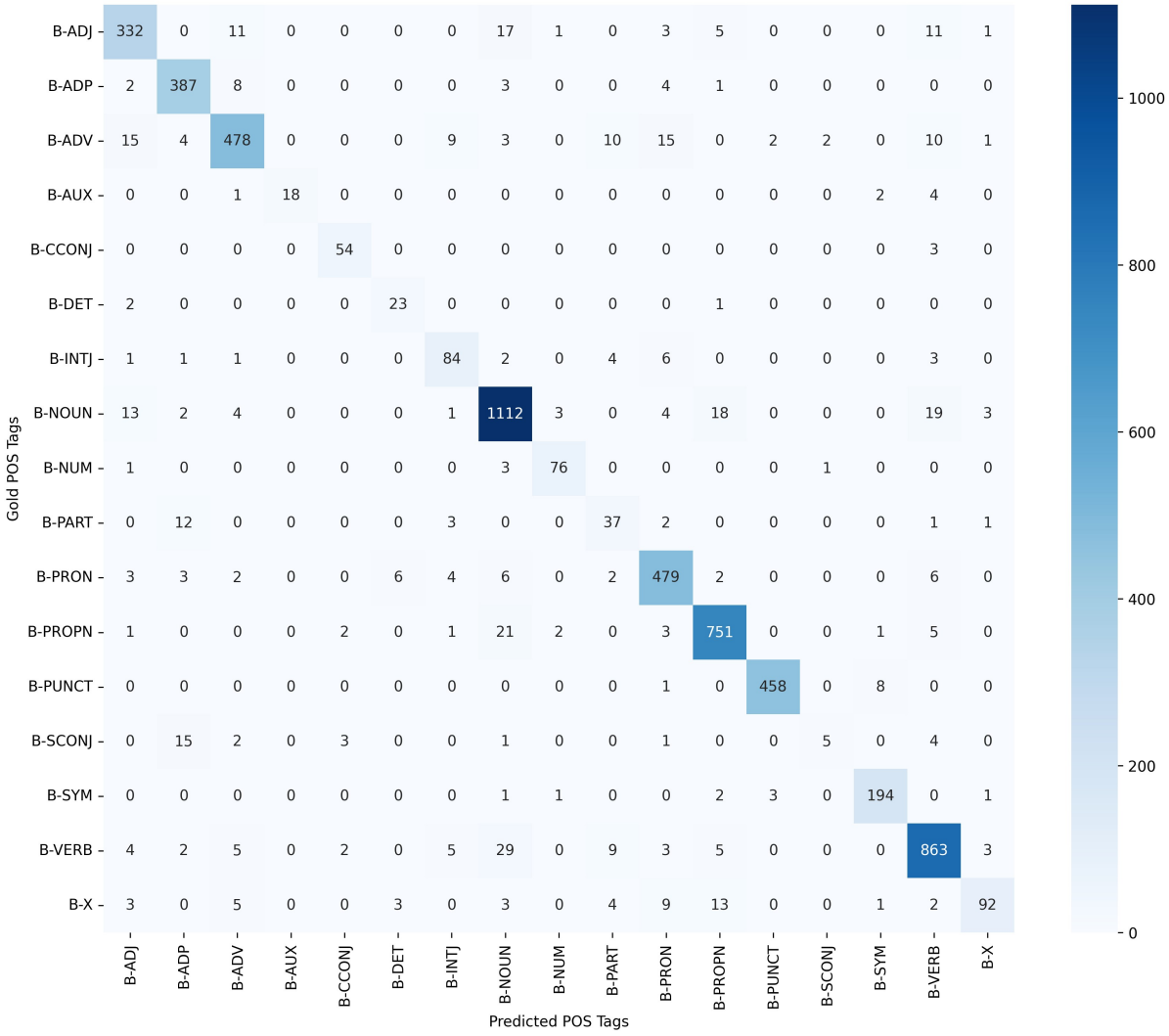

Figure 3: Confusion matrix of predicted vs. gold POS tags for the ICON dataset.