Universal Dependencies for Sindhi

John Bauer
HAI
Stanford University
horatio@cs.stanford.edu

Sakiina Shah
Muhammad Shaheer
Mir Afza Ahmed Talpur
Zubair Sanjrani
Isra University
sakiinashah77@gmail.com
shaheer.memon,afzal.talpur,zubair.sanjrani@isra.edu.pk

Sarwat Qureshi
U. Manchester
sarwatqureshi4@gmail.com

Shafi Pirzada
MLtwist
shafi.pirzada@gmail.com

Christopher D. Manning
Linguistics & Computer Science
Stanford University
manning@stanford.edu

Mutee U Rahman
Isra University
muteeurahman@gmail.com

Abstract

Sindhi is an Indo-Aryan language spoken primarily in Pakistan and India by about 40 million people. Despite this extensive use, it is a low-resource language for NLP tasks, with few datasets or pretrained embeddings available. In this work, we explore linguistic challenges for annotating Sindhi in the UD paradigm, such as language-specific analysis of adpositions and verb forms. We use this analysis to present a newly annotated dependency treebank for Universal Dependencies, along with pretrained embeddings and an annotation pipeline specifically for Sindhi annotation.

1 Introduction

Developing a Universal Dependencies (UD) Treebank for Sindhi presents unique challenges due to the language's complicated linguistic features. Despite being spoken by approximately 40 million people in Pakistan and India, Sindhi is a low-resource language compared to other languages of similar population, with only a few tools and datasets available (see section 2).

Sindhi is a split-ergative Indo-Aryan language, written right to left.[1] It is closely related to Punjabi and Saraiki. Both languages have UD datasets as works in progress (Arora, 2022; Alam et al., 2024), but they are not yet publicly released, and many related languages are also low resource, meaning difficult issues in Sindhi annotation have few references from which to work.

Annotating Sindhi for the increasingly widespread Universal Dependencies framework (de Marneffe et al., 2021) is complicated by its complex case system, where case markers themselves undergo further inflection, alongside rich inflectional morphology with nominal and verbal elements. Additional complexities arise from pronominal suffixation with nouns, postpositions, adverbs, and verbs, and a partially free word order that introduces syntactic ambiguity. Furthermore, the language employs intensifiers across various word classes. The multifunctional use of pronouns as determiners also poses challenges for establishing consistent dependency relations. These characteristics require a tailored approach to create an accurate and robust Sindhi UD treebank, with some of the more difficult topics explained in section 4.

To annotate plain text, current annotation pipelines almost universally use neural methods with static or contextual embeddings. As a low-resource language, Sindhi is most commonly represented in such embeddings as part of a multilingual collection, such as the FastText project (Grave et al., 2018), or as part of a collection of related languages (Khanuja et al., 2021). However, embeddings specifically trained for the language in use typically produce better results, a result this work demonstrates applies to Sindhi. In order to train new embeddings, we collect data from a variety of text sources, described in section 5.

We use the annotated dependency treebank and newly trained word embeddings to train an annotation pipeline using the Stanza package (Qi et al., 2020). To facilitate annotation, we build pipelines from partially annotated data and use these labels as silver annotations for the annotators to use as a baseline.

The main contributions of this work are:

[1] As Sindhi text is written RtL, annotated trees in this text are written RtL, including the English glosses

Proceedings of the Eighth Workshop on Universal Dependencies (UDW, SyntaxFest 2025), pages 105–118
August 27, 2025 ©2025 Association for Computational Linguistics

- build a UD dataset for Sindhi, to be available at universaldependencies.org

- use a multilingual transformer to build progressively better crosslingual models, allowing for quicker annotation

- build Sindhi word vectors and transformers, to be available at huggingface.com

- combine these into a Sindhi annotation pipeline, to be integrated into Stanza at stanza.github.io.

2 Related Work

There are existing datasets for building components of Sindhi annotation pipelines, although none of them are complete solutions.

A partially complete analysis of 665 sentences for Sindhi, named Mazhar Dootio (MD) after its primary author, is available at universaldependencies.org (Dootio and Wagan, 2019). This dataset is small and lacks dependency trees. Nevertheless, it provides a useful baseline, especially for features and lemmatization. In this work, we reanalyze these sentences, adding dependencies and updating the tagging and featurization standards.

Ali et al. (2020b) presents an NER dataset for Sindhi, with 26,000 tagged sentences, available on github.[2] We use it here to build an NER model for the annotation pipeline. The same group also produced a POS dataset of 293K words (Ali et al., 2021).

As static and contextual word embeddings are essential for training neural models, there have been multiple works which provide embeddings. For static embeddings, FastText included Sindhi vectors trained on Common Crawl and Wikipedia (Grave et al., 2018). Later, UESTC crawled a larger corpus of 61M words to produce Sindhi-specific word vectors (Ali et al., 2020a).

For contextual embeddings, XLM-R (Conneau et al., 2020) uses 40M tokens of Sindhi text as part of its training collection. Muril-Large (Khanuja et al., 2021) uses the same amount of Sindhi text, but perhaps because it is focused on Indic and Dravidian languages, tends to perform better on downstream tasks.

Prior work aside from the Mazhar-Dootio dataset also addresses lemmatization, including Nathani et al. (2020) and Dootio and Wagan (2017). We use a different lemmatization standard in this work, though; see section 3.2.

This work uses Stanza (Qi et al., 2020) to build and evaluate models. Similar projects which could build a Sindhi annotation pipeline using the data include UDPipe (Straka, 2018) and spaCy (Honnibal et al., 2020).

3 Annotation Process

The UD dataset described here was annotated in partnership with MLtwist, an NLP annotation company. MLtwist recruited two native speakers of Sindhi for annotation, listed here as authors.

For annotation platforms, we used Datasaur for dependencies and UPOS, followed by Kili for features and XPOS. Individual sentences that needed revision were edited via the free tool conllueditor (Heinecke, 2019), with batch edits executed with Semgrex and Ssurgeon (Bauer et al., 2023).

3.1 Incremental Models

Annotating treebanks from the starting point of a silver dataset is a common practice. The English EWT dependency treebank (Silveira et al., 2014) started with LDC constituency trees using a deterministic conversion process (Manning et al., 2014). The English GUM dependency treebank (Zeldes, 2017) uses a parser trained on existing English materials to provide initial silver trees to the annotators. As shown in Mikulová et al. (2022), using silver trees improves the final annotation quality.

Meanwhile, using multilingual representations to improve parsers is a common practice, such as in recent Javanese parsing work (Ghiffari et al., 2024), which used two layers of multilingual representations to build the parser.

This work combines those two concepts to build initial silver trees, even in the setting of very few trees already annotated.

The Muril transformer (Khanuja et al., 2021) contains all of the scheduled languages in India, of which Sindhi is one. Several languages in Muril are also represented in Universal Dependencies, shown in table 1:

Language	Family	Total Trees
Hindi	Indic	16,649
Urdu	Indic	5,130
Marathi	Indic	466
Tamil	Dravidian	600

Table 1: Languages in Muril with sufficient UD data and the size of the largest treebank

[2] https://github.com/AliWazir/SiNER-dataset

To build silver trees, we train a Stanza POS tagger and dependency parser with a mix of trees from these languages, even without having labeled Sindhi dependency trees available.[3]

To demonstrate the effectiveness of this approach, we train models based on subsets of the available data and score them on the Sindhi test set. Each model trained here includes 1000 Urdu trees, 1000 Hindi trees (Bhat et al., 2017; Palmer et al., 2009), 373 Marathi trees (Ravishankar, 2017), and 400 Tamil trees (Ramasamy and Žabokrtský, 2012).[4] Tamil is an interesting case, as it is Dravidian rather than Indic, but we find there is still some benefit to transfer learning for dependencies. Table 2 shows that even low amounts of gold Sindhi data result in high-quality silver trees, when combined with multilingual training.

Train	Dev	None	Indic	Full
		POS AllTags F1		
0	0	–	82.66	81.94
0	100	–	81.72	82.22
50	100	85.51	87.15	86.88
100	100	89.81	89.39	89.54
200	500	90.65	91.06	90.91
500	500	92.85	92.64	90.90
		Depparse LAS F1		
0	0	–	69.40	72.91
0	100	–	72.02	72.52
50	100	61.54	75.13	76.01
100	100	70.38	78.71	78.74
200	500	76.73	80.87	80.88
500	500	81.33	82.68	82.53

Table 2: Accuracy of crosslingual models. Annotation gets easier when adding more data. "Indic" means the model is trained with additional Indic UD data, and "Full" includes Tamil as well.

3.2 Lemmatization

For the most part, we follow the Dootio and Wagan (2019) standard for lemmatization, with one key difference. In the former, the lemma is chosen to be the root form of the word regardless of POS tag, whereas here the lemma uses the default root form with the same POS tag. We remove inflections, but not derivational endings. For example, in the English dataset EWT (Silveira et al., 2014), *ADJ* words ending with *-ish* are lemmatized to the *-ish* form, whereas the Mazhar Dootio standard would stem the word without the *-ish*.

Form	w/o POS	w/ POS
hawkish	hawk	hawkish
childish	child	childish
sympathize	sympathy	sympathize

The results in Sindhi follow a similar pattern:

Sindhi ADJ	M-D	Ours
اايماندار	ايمان	ايماندار
ٿڌي	ٿڌ	ٿڌو
دولتمند	دولت	دولتمند

Repeated iteration of training a Stanza seq2seq lemmatizer helped speed the lemmatization process, as once 1000 lemmas were manually lemmatized, the lemmatizer model demonstrated an 86% accuracy on unseen lemmas. In contrast, the "identity" lemmatizer, using the word itself, would only be 66% accurate on unseen words.

3.3 Corpus Statistics

After retokenizing and deduplicating the MD dataset, 509 annotated sentences remain. We use this and a collection of 378 sentences of news articles and folk tales to build the test set. Another 386 sentences will comprise the dev set. As of this writing, 4116 total sentences have XPOS and features annotated. Another 1532 have UPOS and dependencies and will be fully annotated by the time of publication.

3.4 Agreement

To measure interannotator agreement, we gave the same batch of 200 silver sentences to both of the UPOS and dependencies annotators. The resulting annotations agreed on 96% of the UPOS (95.4 kappa) and 91.4% of the dependencies.

These numbers have a caveat, though, in that agreement will be much higher when annotating silver trees, as the annotators will be predisposed to choose the silver label in the case of an ambiguous annotation. This phenomenon has been previously studied in work such as Berzak et al. (2016).

4 Linguistically adapting UD for Sindhi

Several constructions in Sindhi are challenging to annotate as part of a Universal Dependencies framework, or are seen only in the context of other low-resource languages.

4.1 Demonstratives as Determiners

Similar to many Indo-Aryan languages, and others such as the Slavic language Czech, Sindhi lacks standalone articles and does not have a well-defined category of determiners. Instead, demonstrative pronouns often function as determiners.

[3] With 0 Sindhi trees, the Urdu dev set is used.

[4] The entire training set for Marathi and Tamil, and a random subset of Hindi and Urdu

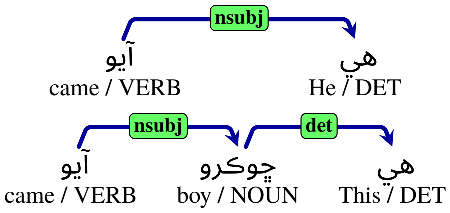
Therefore, Sindhi demonstrative pronouns map to the *DET* part of speech.

This multifunctional use of demonstratives can lead to ambiguity and requires careful annotation in Universal Dependencies relations. For example, هي hī can be a demonstrative (proximal *he*) or a determiner (*this*). The use of هي as both a demonstrative and a determiner can be seen in the following sentences:

1. هي آيو. hī āyo (He came)
2. هي چوڪرو آيو. hī chokro āyo (This boy came)

In the first sentence, هي is a demonstrative pronoun used standalone and functions as the nsubj of the sentence. In the second sentence, هي is used as a determiner modifying چوڪرو, where چوڪرو is the nsubj. When used standalone in a sentence, demonstratives function as pronouns and can appear in the roles of nsubj, obj, iobj, or obl. However, when used before nominals, they function as determiners.

As argued in (Bharati et al., 2008), it can be useful to have a separate part of speech for demonstratives compared to other pronouns. Furthermore, the treebanks for other Indic languages such as Hindi and Urdu establish the convention of always tagging demonstratives *DET*. It is useful to have this distinction even when they function as pronouns, even though this contrasts with English UD, which contextually distinguishes "this" as *PRON* or *DET*, for example. In such cases, the dependency relation demonstrates the use of the word.



4.2 Intensifiers

One common class of words in Sindhi is *intensifiers*. These are words with no direct English translation which put the emphasis of the sentence on the previous word. In English writing, bold or italic text would have a similar effect.

An example sentence from this dataset is:

گهرجي	ڪوٺڻ	ادب	ٻاراڻو	ئي	کي	ان
ghurje	kōṭhaṇ	adab	bārāṇo	ī	khe	un
need	calling	lit.	childish	EMP	case	that

It should be called children's literature

In this sentence, ئي is an intensifier. Figure 1 has an analysis of the phrase ان کي ائي.

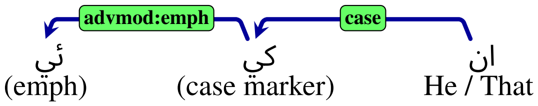


Figure 1: The intensifier ئي turns "he" into "Only he"

In Sindhi, intensifiers are typically labeled with the part-of-speech tag *PART* having *advmod:emph* relation, attached to the modified word.

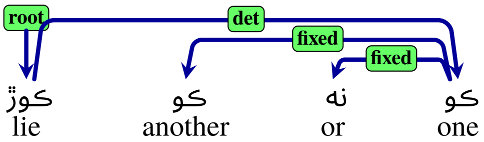
Intensifier words may not always act as intensifiers and can serve other roles based on the context. Notably, the particle نه (na) is usually a negation particle, annotated as *PART* with the *advmod* relation, but it functions as an intensifier when following imperative verbs, as in اچ نه (ach na, "come on"), where it emphasizes the command. However, when نه precedes the verb, as in نه اچ (na ach, "don't come"), it acts as a negation particle. Similarly, ته (ta) is generally a subordinating conjunction, annotated as *SCONJ* with the *mark* relation, but it becomes an intensifier when following specific verb forms, as in اچ ته (ach ta, "come on"), or nominals, as in هو ته (hu ta, "he too" or "he indeed"). These multifunctional particles highlight the need for context-sensitive annotation in Sindhi UD.

4.3 Fixed

In Universal Dependencies (UD), the *fixed* relation connects words within a multiword expression (MWE) where the fixed combination functions as a single grammatical unit with non-compositional meaning. For example, ڪو نه (ko na, "not any") is a two-word fixed expression annotated with the *advmod* relation, representing negation. Individually, ڪو (ko) can serve as a determiner (*det*) or an unspecified dependency (*dep*), while نه (na) acts as a negation particle or intensifier (typically *PART*, see section 4.2). Together, however, these two tokens form a unit with an *advmod* role. Because of their rigidity and role as single grammatical units, we label these *fixed*.

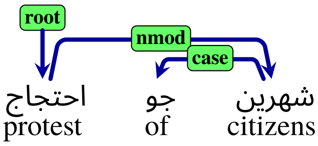
Some examples include:

ڪو نه	not any	advmod
ڪو نه ڪو	one or another	det
کان سواءِ	except	case
نه ئي	nor	advmod



4.4 Postpositions

Most adpositions in Sindhi, tagged *ADP* as per UD standards, are postpositions. An entire adpositional phrase typically precedes the modified head. A simple example in this dataset is "The protest of the citizens":



Postpositions in Sindhi frequently function as case markers, appearing after nominals and requiring those nominals to take the oblique case form. This oblique case is a fundamental prerequisite for further case marking in the language. Consider the following examples:

هي گاڏيون (hī gāḍiyūn) "These vehicles"

هنن گاڏين تي (hinan gāḍīan te) "On these vehicles"

In the first example, both elements of the adpositional phrase " هي" (hī, "these") and " گاڏيون" (gāḍiyūn, "vehicles") are in the default nominative (or direct) case, as no postposition is present. In the second example, the postposition " تي" (te, "on") triggers the oblique case, causing both the determiner " هي" (hī) to inflect to " هنن" (hinan) and the noun " گاڏيون" (gāḍiyūn) to inflect to " گاڏين" (gāḍīan). This illustrates how postpositions in Sindhi govern the case of the preceding nominals, shifting them from nominative to oblique forms (accusative case in terms of Universal Dependencies).

Sindhi genitive adpositions (ADPs) differ from other ADPs, such as locative ones, due to their ability to inflect for number, gender, and case. Unlike locative ADPs like " ۾" (mē, "in"), " تي" (te, "on"), and " وٽ" (vat, "near/beside"), which only mark the accusative (oblique) case of preceding nominals and lack further inflectional features, genitive ADPs not only assign case to the nouns they follow but also bear their own inflectional features.

Examples of the distinctions between these features can be found in appendix B.

We use XPOS to separate out two classes of adpositions, genitive and location. Genitive ADP are inflected for case and are featurized for nominative and accusative cases. ADP that represent location are treated as having no further features, as are ADP which are not one of these two subclasses. We do not use *Case=Loc* for location ADP as there is no inflection which indicates the ADP is locative.

4.5 Prepositions

There are some rare cases of prepositions, such as the word سواءِ, "except", in the sentence analyzed in figure 2:

جنهن سان سواءِ جاهلن جي هر هڪ جو تعلق آهي

4.6 Participles and VerbForm Features

Participles in Sindhi are versatile verb forms that function as adjectives, nouns, and adverbials. They also play a role in tense and aspect formation. Participles in Sindhi include verbal nouns and the present, past, future, and conjunctive participles. Participles are further marked by number, gender, and person inflections, encoded in the features. These features, along with the *VerbForm* feature, distinguish the different participles.

Examples features for the different verb forms described here are in table 6.

Present Participle The present participle in Sindhi is marked by the verbal suffix " ند" (and), further inflected for number and gender. The following sentence is an example of a present participle paired with an auxiliary verb forming imperfective habitual aspect in present tense where the main verb لک (likh, "write") is inflected by masculine singular suffix " ندو" (ando), forming the present participle لکندو (likhando, habitual "writes").

هو لکندو آهي (hu likhando āhe, "He writes")

Past Participle The past participle in Sindhi is marked by the suffix " يل" (yal), which rarely inflects for number (e.g., " لکيلن" /likhyalan/ for plural in some dialects). In the following sentence, the verb لک (likh, "write") with the suffix " يل" (yal) forms the past participle form لکيل (likhyal, "written"), with perfective aspect in passive voice.

هي ڪتاب لکيل آهي hī kitāb likhil āhe "This book is written"

Future Participle The future participle in Sindhi is often an inflected form of the infinitive. For example, the infinitive " لکڻ" (likhaṇ, "to write"), is marked by number and gender to indicate obligation or futurity. In the following sentence, the verb form لکڻو (likhaṇo, "to write" or "having to write") is a masculine singular inflection of the above infinitive with an imperfective aspect.

مون کي ڪتاب لکڻو آهي (mūn khē kitāb likhaṇo āhe, "I have to write a book")

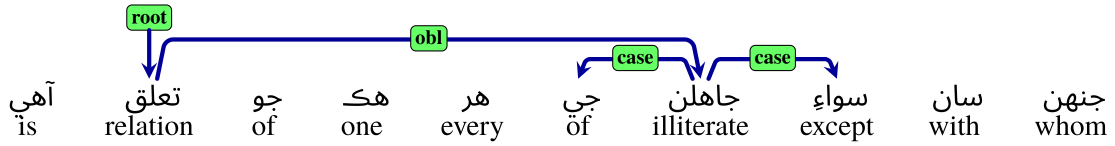


Figure 2: Very rarely, Sindhi has prepositions as well as postpositions
"To which everyone except the illiterate belongs"

Unlike the present and past participles where *VerbForm=Part* is used, we label this *VerbForm=Inf* to avoid ambiguity.

Conjunctive Participle The conjunctive participle in Sindhi typically denotes an action completed before the main verb or clause's action, linking two events without an explicit conjunction. It is formed by inflecting the verb root with the suffix " ي" , as in لکي (likhi) from لک (likh, "write"). However, this form's role, either as a simple verb or a conjunctive participle, depends not only on its morphology but also on its syntactic standing in the sentence. In the following examples, syntactic context such as adverbial modification of the main verb distinguishes two potential roles of لکي (likhi):

A simple past verb: لکي چٺي مون (mūn chithī likhi, "I wrote a letter"). Here, لکي is the root.

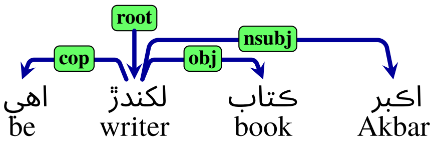
A conjunctive participle: آيس گهر لکي چٺي آئون (āūn chithī likhi ghar āyas, "I came home after writing a letter"). Here, it modifies the root آيس (āyas, "came"), indicating a prior action.

Conjunctive participles in Sindhi do not inflect for number, gender, or person; they are invariable, unlike the present or past participles discussed above. Their aspect is always perfective, reflecting a completed action relative to the main clause.

In Universal Dependencies, the conjunctive participle is mapped to the converb category, tagged with *VerbForm=Conv*. Across UD, a converb is a verb form that adverbially modifies the main verb, often indicating sequence, cause, or manner.

Verbal Noun Verbal nouns in Sindhi are derived from verbs and function as nouns while retaining verbal properties like implying an action. Unlike infinitive forms like لکڻ (likhaṇ, "to write"), which can also act as nouns, Sindhi has a distinct class of verbal nouns marked by the agentive suffix ندڙ (ndar), indicating the doer (e.g., لک (likh, "write") becomes لکندڙ (likhandar, "writer")).

In the sentence اکبر ڪتاب لکندڙ آهي (Akbar kitāb likhandar āhe, "Akbar is the writer of the book"), لکندڙ (likhandar) is a verbal noun, acting as a predicate noun with imperfective aspect (ongoing capacity to write) and singular number. It inflects for number and case, as in لکندڙن کي ٻڌايو ڪتاب (kitāb likhandaran khē buḍhāyo, "Tell the writers of the book"), where لکندڙن (likhandaran) is plural, marked with accusative case via کي (khē, "to").



In the dependency parse of ڪتاب لکندڙ آهي اکبر, there is no *ADP* indicating "of the book", but rather "book" is the *obj* of "writer". Furthermore, unlike English deverbals such as "chaser" in "ambulance chaser", this potential *obj* can have many substitutions, including complete phrases. As "writer" is taking on VERB dependencies, we follow the analysis of Cecchini (2021) and an extensive UD discussion[5] and treat these as *VERB*, with the feature *VerbForm=Vnoun*.

Infinitive Infinitive verbs in Sindhi are formed by the ڻ (-ṇ) suffix inflection. For example, لک (likh) 'write' becomes لکڻ (likhaṇ) "to write". Infinitives always exhibit the imperfective aspect and are marked by the feature *VerbForm=Inf*.

The lack of further inflection distinguishes infinitives from future participles. Although both share the *VerbForm=Inf* and *Aspect=Inf* features, they are distinguished by the remaining features. Unlike infinitives, future participles inflect for number, gender, and person.

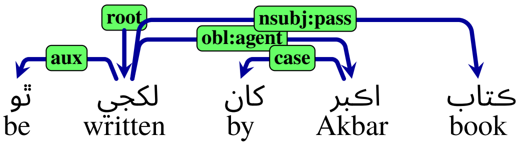
Infinitives can appear in various syntactic roles within the Universal Dependencies (UD) structure, including *nsubj* (nominal subject), *obj* (object), *xcomp* (open clausal complement), *ccomp* (clausal complement), and *advcl* (adverbial clause).

4.7 Passive Voice

Sindhi forms passive voice constructions across various tenses and aspects, primarily utilizing mor-

[5] https://github.com/UniversalDependencies/docs/issues/1125

phological passive forms and auxiliary verbs. In the present tense with imperfective aspect, passive forms are marked by suffixes such as “ جي” (-je) or “ بو” (-bo), inflected for number and gender, and paired with present tense auxiliaries like “ ٿو” (tho) or “ آهي” (āhe). For example, the active sentence “ اکبر کتاب لکي ٿو” (Akbar kitāb likhe tho, “Akbar writes a book”) becomes “ کتاب اکبر کان لکجي ٿو ” (kitāb Akbar kān likhje tho, “The book is written by Akbar”) for masculine singular. Transition from active to passive voice cause a usual role shift like subject to oblique agent and object to passive subject marked by the UD relations *obl:agent* and *nsubj:pass* respectively. However, the auxiliary is not necessarily tagged with *aux:pass* when the morphological form of the main verb inherently indicates passivity as it serves solely to mark tense and is thus tagged with the general *aux* relation. This can be seen in the following example:



Similarly, “کتاب اکبر کان لکبو آهي” (kitāb Akbar kān likhbo āhe, “The book is written by Akbar”) and “چٺيون لکبيون آهن” (chithiyūn likhbiyūn āhan, “Letters are written”) show passive constructions by using the “ بو” (-bo) suffix, which varies by gender and number (e.g., “ لکبي” /likhbī/ for feminine singular).

As is true in many languages, including English, it is possible to use these constructions for each of past, present, and future tense. For further discussion of how these verbs inflect, see appendix C.

4.8 Non-projectivity

Sindhi is a highly projective language, with less than 6% of the sentences in the dataset having non-projective arcs. There are some cases where this occurs, though. An example non-projective sentence is given in figure 3.

4.9 Pronominal Suffixes

Sindhi employs pronominal suffixes on verbs, nouns, and postpositions to encode possessive, subject, or object roles, as in “ لکيومانس” (likhyomāns, “I wrote to him”). These suffixes blur the line between affixes (bound morphemes) and clitics (semi-independent), posing challenges in UD annotation for tokenization, UPOS tagging, and dependency relations.

To analyze these tokens, two approaches are used across UD datasets.

Single Token Approach Treat the word as one token with rich morphological features capturing the host’s POS (e.g., *VERB*) and suffix roles (e.g., *Person[Subj]*=1, *Person[iObj]*=3).

This can capture the morphology, but it obscures syntactic relations (e.g., ambiguity between object, indirect object, or oblique roles).

Split Token Approach Split suffixes into separate tokens (e.g., *PRON*), each with its own features and relations (e.g., *nsubj*, *iobj*).

This clarifies the syntax via relations. However, it risks ambiguity, as suffixes like “ س” (-s) vary by context (e.g., 3rd singular in مارينس” /mārīns/, “he was beaten”, vs. 1st singular in “ هئس” /hēs/, “I was”).

UD treebanks adopt either approach based on language-specific arguments and conventions, favoring single tokens for bound suffixes but splitting clitic-like forms. Challenges include assigning UPOS (*VERB* vs. *PRON*), defining relations, and resolving ambiguities in suffix meaning.

Here, we choose to split these suffixes using the multi-word token approach, with a compelling reason being that some of the pronouns themselves inflect with features such as *Case*, which are not adequately represented in the single token analysis.

5 Embedding Pretraining

As has been typical in the last decade of NLP work, before building the annotation pipeline, we pretrained static word embeddings, a character model (Akbik et al., 2018), and a transformer using a corpus of raw text. The corpus collected is 372M words, significantly less than the original BERT, which used 3,300M words for English (Devlin et al., 2019), but sufficient for pretraining models for a new language. This is significantly larger than that used in Muril and XLM-R, each with less than 40M.

Source	M Tokens	
Wikipedia	5	
Oscar CC	11	(Abadji et al., 2022)
Books Corpus	2	(Ali et al., 2020a)
Adabi Forums	0.5	
Sindhi LA Journals	0.8	
Sindhi LA Encyclopedia	5	
Various newswire	38	
Sindh Salamt Forums	24	
Sindh Salamt Books	57	
Sangraha	229	(Khan et al., 2024)

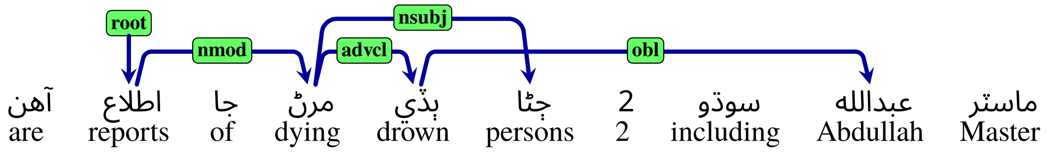


Figure 3: In this non-projective tree, عبدالله attaches to ٻڏي despite the intervening subject "Two people, including Master Abdullah, are reported to have drowned."

We experimented with GloVe (Pennington et al., 2014), FastText (Grave et al., 2018), and word2vec (Mikolov et al., 2013), the latter two using the Gensim implementation (Rehurek and Sojka, 2011).

Model	NER Entity F1	Tags F1	Depparse LAS
Original	84.25	76.10	80.92
GloVe	86.07	77.90	81.69
Word2Vec	85.22	77.24	81.96
FastText	82.12	77.09	81.75
GloVe w/ CharLM	86.11	80.90	82.84

Table 3: Dev scores for three annotators using static word embeddings, potentially with charlm

Based on these results, we use the GloVe model, to be distributed with the Stanza pipeline for Sindhi. Other pretrained models will be available along with the default Stanza pipeline.

Stanza (Qi et al., 2020) implements a contextual character model. Table 3 also demonstrates the benefits of this model.

We used HuggingFace's Roberta implementation to finetune Muril and to train two candidate Sindhi transformers from a random initialization. Interestingly, and unfortunately, each of the models produced were less accurate on the three tasks compared with the original Muril-Large (see table 4). Possible explanations include that Muril incorporated an additional machine translation learning objective and a large multilingual corpus. Future work will involve experimenting with multilingual datasets with the larger Sindhi collection or with a more targeted low-resource technique such as MicroBERT (Gessler and Zeldes, 2022).

Model	NER Entity F1	Tags F1	Depparse LAS
Muril	87.06	78.03	84.30
Finetuned	86.78	80.10	83.13
6 layers	82.66	75.59	81.07
12 layers	81.75	76.31	81.77

Table 4: Dev scores for three annotators using transformers

6 Annotation Pipeline

Having annotated a UD dataset and built some pretrained models, we then built a pipeline using Stanza annotation software. For NER, we used the existing SiNER dataset (Ali et al., 2020b).

Lemma and NER scores are high. Surprisingly, feature tagging is presently the lowest, performing below dependency and NER. Continued improvements to the annotations and the embeddings should improve scores.

Task	Emb	Score
Lemma	seq2seq w/ charlm	99.08 Accuracy
UPOS	Muril	98.44 F1
Features	Muril	85.56 F1
Depparse	Muril	91.83 LAS
NER	Muril	88.13 Entity F1

Table 5: Test scores for 5 Sindhi tasks

For a breakdown of individual relations, see Appendix E.

7 Conclusion

This project explores the more challenging structures for analyzing Sindhi grammar using Universal Dependencies. We then analyze roughly 6000 sentences of Sindhi, building a dataset for release in the 2.16 and 2.17 releases of UD. In parallel, we collect a larger corpus of Sindhi text than previously collected, facilitating more accurate static embeddings for the language. Combining each of these tasks with Stanza results in the first end-to-end annotation pipeline for the Sindhi language.

Limitations

Stanza includes three additional annotators, currently not implemented. A coreference dataset would be a useful addition to Sindhi annotation, but was beyond the scope of this work. As noted in section 2, there are sentiment datasets published for Sindhi, but as of this writing none have been made publicly available. There is also a constituency parser, but given the recent trend towards dependency parsing instead of constituency parsing, it is unlikely there will be a gold annotation of this treebank for constituencies.

The dataset utilised in this study has several limitations that may impact the reliability of our findings. The data primarily consisted of text from the Kawish newspaper. Many of the sentences are headlines, resulting in significant syntactical shortcomings, as these brief statements often highlighted named entities without forming complex linguistic structures. Key syntactical elements were largely absent due to newspaper headlines' inherent brevity and formulaic nature.

Furthermore, the dataset showed a significant lack of tense variation, with an overrepresentation of present-tense constructions to convey immediate facts. In contrast, past and future tense forms were either underrepresented or completely absent. This lack of tense diversity hindered capturing important temporal variations essential for comprehensive language modeling, resulting in limited morphological variation in our model.

Additionally, the repeated presence of named entities, geographical references, and thematic content typical in news publications generated potential biases in the distributional properties of the language representation. To improve future iterations of this research, a more diverse Sindhi language corpus should be developed using high-quality sources, including:

- **Sindh Textbook Board**[6]. Provides academic and domain-specific linguistic material.
- **Sindhi Adabi Board**[7]. Offers access to classical and modern Sindhi literature.
- **Encyclopedia Sindhiana**[8]. A rich source for definitional and encyclopedic language patterns.
- **Literary works of various authors**, such as Anwer Pirzado. Contributes complex syntactic structures and culturally rich narratives.

Incorporating these resources will enhance linguistic richness, expand tense usage, reduce redundancy, and increase both lexical and syntactic diversity—ultimately supporting the development of more robust and representative Sindhi language models.

Ethics Statement

As an annotation project for a language primarily spoken in Pakistan and India, with two non-local authors on the author list, the primary ethical concern for this paper would be that of "parachute research", as described in Odeny and Bosurgi (2022) for the medical domain.

To ensure fairness in the process, authorship credits were offered to each of the annotators. Furthermore, Professor Rahman, leading the research group at ISRA, occupies the lead author position in the author list.

As justification for the inclusion of the non-local authors, one provided extensive linguistics advice, whereas the other contributed significant technical time in terms of aligning the treebank with UD requirements, building models, and submitting pull requests and/or issues to conllueditor, Stanza, and Semgrex for features needed to support this project.

In terms of research relevant to the local community, one of the authors of the paper is a native speaker who started the project out of a desire to see this type of annotation pipeline available for Sindhi.

It is our hope that these practices will not only call attention to the strong linguistics group at Isra University, but also facilitate further collaborations between "Global North" research groups and local research groups.

Acknowledgments

We would like to thank the members of the UD community for their efforts advising the annotation process. We also appreciate the valuable feedback from our SyntaxFest reviewers. MLtwist put in a lot of work without which this project would not have been possible. Finally, we thank the licensees of CoreNLP for their continued support of StanfordNLP, which provided funding for this work.

References

Julien Abadji, Pedro Ortiz Suarez, Laurent Romary, and Benoît Sagot. 2022. Towards a Cleaner Document-Oriented Multilingual Crawled Corpus. *arXiv e-prints*, arXiv:2201.06642.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–

[6] https://stbb.edu.pk

[7] https://sindhiadabiboard.net/library/

[8] https://encyclopediasindhiana.org/

1649, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Meesum Alam, Francis Tyers, Emily Hanink, and Sandra Kübler. 2024. Universal Dependencies for Saraiki. In *Proceedings of the Joint Workshop on Multiword Expressions and Universal Dependencies (MWE-UD) @ LREC-COLING 2024*, pages 188–197, Torino, Italia. ELRA and ICCL.

Wazir Ali, Jay Kumar, Junyu Lu, and Zenglin Xu. 2020a. Word embedding based new corpus for low-resourced language: Sindhi. *Preprint*, arXiv:1911.12579.

Wazir Ali, Junyu Lu, and Zenglin Xu. 2020b. SiNER: A large dataset for Sindhi named entity recognition. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2953–2961, Marseille, France. European Language Resources Association.

Wazir Ali, Zenglin Xu, and Jay Kumar. 2021. SiPOS: A benchmark dataset for Sindhi part-of-speech tagging. In *Proceedings of the Student Research Workshop Associated with RANLP 2021*, pages 22–30, Online. INCOMA Ltd.

Aryaman Arora. 2022. Universal Dependencies for Punjabi. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5705–5711, Marseille, France. European Language Resources Association.

John Bauer, Chloé Kiddon, Eric Yeh, Alex Shan, and Christopher D. Manning. 2023. Semgrex and ssurgeon, searching and manipulating dependency graphs. In *Proceedings of the 21st International Workshop on Treebanks and Linguistic Theories (TLT, GURT/SyntaxFest 2023)*, pages 67–73, Washington, D.C. Association for Computational Linguistics.

Yevgeni Berzak, Jessica Kenney, Carolyn Spadine, Jing Xian Wang, Lucia Lam, Keiko Sophie Mori, Sebastian Garza, and Boris Katz. 2016. Universal Dependencies for learner English. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 737–746, Berlin, Germany. Association for Computational Linguistics.

Akshar Bharati, Dipti Misra Sharma, Lakshmi Bai, and Rajeev Sangal. 2008. Anncorra : Annotating corpora guidelines for pos and chunk annotation for Indian languages. Technical report, Language Technologies Research Center, IIIT Hyderabad.

Riyaz Ahmad Bhat, Rajesh Bhatt, Annahita Farudi, Prescott Klassen, Bhuvana Narasimhan, Martha Palmer, Owen Rambow, Dipti Misra Sharma, Ashwini Vaidya, Sri Ramagurumurthy Vishnu, and 1 others. 2017. The hindi/urdu treebank project. In *Handbook of Linguistic Annotation*. Springer Press.

Flavio Massimiliano Cecchini. 2021. Formae reformandae: for a reorganisation of verb form annotation in Universal Dependencies illustrated by the specific case of Latin. In *Proceedings of the Fifth Workshop on Universal Dependencies (UDW, SyntaxFest 2021)*, pages 1–15, Sofia, Bulgaria. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. *Preprint*, arXiv:1911.02116.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Mazhar Dootio and Asim Wagan. 2017. Automatic stemming and lemmatization process for Sindhi text. *JSSIR*, 6:19–28.

Mazhar Ali Dootio and Asim Imdad Wagan. 2019. Syntactic parsing and supervised analysis of Sindhi text. *Journal of King Saud University – Computer and Information Sciences*, 31(1):105–112.

Luke Gessler and Amir Zeldes. 2022. MicroBERT: Effective training of low-resource monolingual BERTs through parameter reduction and multitask learning. In *Proceedings of the 2nd Workshop on Multi-lingual Representation Learning (MRL)*, pages 86–99, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Fadli Aulawi Al Ghiffari, Ika Alfina, and Kurniawati Azizah. 2024. Cross-lingual transfer learning for Javanese dependency parsing. *Preprint*, arXiv:2401.12072.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Johannes Heinecke. 2019. ConlluEditor: a fully graphical editor for universal dependencies treebank files. In *Proceedings of the Third Workshop on Universal Dependencies (UDW, SyntaxFest 2019)*, pages 87–93, Paris, France. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Mohammed Khan, Priyam Mehta, Ananth Sankar, Umashankar Kumaravelan, Sumanth Doddapaneni, Suriyaprasaad B, Varun G, Sparsh Jain, Anoop Kunchukuttan, Pratyush Kumar, Raj Dabre, and Mitesh Khapra. 2024. IndicLLMSuite: A blueprint for creating pre-training and fine-tuning datasets for Indian languages. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, page 15831–15879. Association for Computational Linguistics.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, Shruti Gupta, Subhash Chandra Bose Gali, Vish Subramanian, and Partha Talukdar. 2021. MuRIL: Multilingual representations for Indian languages. *Preprint*, arXiv:2103.10730.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

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

Marie Mikulová, Milan Straka, Jan Štěpánek, Barbora Štěpánková, and Jan Hajic. 2022. Quality and efficiency of manual annotation: Pre-annotation bias. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2909–2918, Marseille, France. European Language Resources Association.

Bharti Nathani, Nisheeth Joshi, and G.N. Purohit. 2020. Design and development of unsupervised stemmer for Sindhi language. *Procedia Computer Science*, 167:1920–1927. International Conference on Computational Intelligence and Data Science.

Beryne Odeny and Raffaella Bosurgi. 2022. Time to end parachute science. *PLOS Medicine*, 19(9):1–3.

Martha Palmer, Rajesh Bhatt, Bhuvana Narasimhan, Owen Rambow, Dipti Misra Sharma, and Fei Xia. 2009. Hindi syntax: Annotating dependency, lexical predicate-argument structure, and phrase structure. In *The 7th International Conference on Natural Language Processing*, pages 14–17.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

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

Loganathan Ramasamy and Zdeněk Žabokrtský. 2012. Prague dependency style treebank for Tamil. In *Proceedings of Eighth International Conference on Language Resources and Evaluation (LREC 2012)*, pages 1888–1894, İstanbul, Turkey.

Vinit Ravishankar. 2017. A Universal Dependencies treebank for Marathi. In *Proceedings of the 16th International Workshop on Treebanks and Linguistic Theories*, pages 190–200, Prague, Czech Republic.

Radim Rehurek and Petr Sojka. 2011. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2).

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Christopher D. Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*.

Milan Straka. 2018. UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, Brussels, Belgium. Association for Computational Linguistics.

Amir Zeldes. 2017. The GUM corpus: Creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612.

A Features

Table 6 demonstrates the morphological features used for the various verb forms described in section 4.6.

B Postposition Examples

As explained in section 4.4, genitive adpositions in Sindhi inflect to agree with the associated noun, whereas ADPs representing location do not inflect. For example, consider the locative ADPs in these sentences:

- “ مٿي تي ٽوپي” (mathe te topī, “A cap on the head”) the noun “ مٿو” (matho, “head”) shifts to its oblique form “ مٿي” (mathe) before “ تي” (te).

- “ پاڙي ۾ دڪان” (pāṛe mē dukān, “A shop in the neighborhood”) “ پاڙو” (pāṛo, “neighborhood”) becomes “ پاڙي” (pāṛe).

Example	Verb	Verb Type	VerbForm	Aspect	Gender	Number	Other
هو لکندو آهي hu likhando āhe, “He writes”	لکندو	Pres. Part	Part	Imp	Masc	Sing	–
هي ڪتاب لکيل آهي hī kitāb likhil āhe “This book is written”	لکيل	Past Part.	Part	Perf	–	–	Voice=Pass
مون کي ڪتاب لکڻو آهي mūn khē kitāb likhaṇo āhe “I have to write a book”	لکڻو	Fut Part	Inf	Imp	Masc	Sing	–
آئون چٺي لکي گهر آيس āūn chithī likhi ghar āyas “I came home after writing a letter”	لکي	Converb	Conv	Perf	–	–	–
ڪتاب لکندڙن کي ٻڌايو kitāb likhandaran khē buḍhāyo “Tell the writers of the book”	لکندڙن	Verbal Noun	Vnoun	Imp	–	Plur	Case=Acc
ڪتاب لکڻ سٺو آهي kitāb likhaṇ sutho āhe “It is good to write a book”	لکڻ	Infinitive	Inf	Imp	–	–	–

Table 6: Morphological features for example verb phrases

- “ مون وٽ ويھ” (mūn vat vēh, “Sit near me”) “ آئون” (āūn, “I”) shifts to “ مون” (mūn “me”).

In these examples, the locative ADPs mark the accusative/oblique case of the preceding nouns, which inflect accordingly (e.g., “ متو” → “ متي”), but the ADPs themselves remain uninflected and are typically tagged as locative postpositions (PSPL).

In contrast, genitive ADPs, such as “ جو” (jo, “of”), also govern the accusative/oblique case of preceding nominals but are themselves inflected for number, gender, and case, agreeing with the following noun. Consider these examples:

- “ گهوڙي جو پڇ” (ghoṛe jo puchh, “the tail of the horse”) “ گهوڙو” (ghoṛo, “horse”) becomes “ گهوڙي” (ghoṛe, accusative singular masculine), and “ جو” (jo) is nominative singular masculine, matching “ پڇ” (puchh, “tail”).

- “ گهوڙي جا ڪن” (ghoṛe jā kan, “the ears of the horse”) “ جو” inflects to “ جا” (jā, nominative plural masculine) for “ ڪن” (kan, “ears”).

- “ گهوڙي جون اکيون” (ghoṛe jūn akhiyūn, “the eyes of the horse”) “ جو” becomes “ جون” (jūn, nominative plural feminine) for “ اکيون” (akhiyūn, “eyes”).

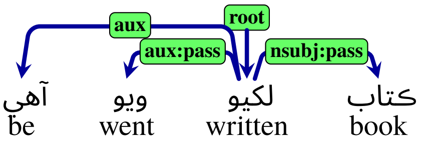
- “ گهوڙي جي پڇ جو وار” (ghoṛe je puchh jo vār, “the hair of the tail of the horse”) “ گهوڙي جي پڇ” (ghoṛe je puchh) uses “ جي” (je, accusative singular masculine) governed by “ پڇ” (puchh, accusative), while “ جو” (jo) is nominative singular masculine for “ وار” (vār, “hair”).

The genitive ADP “ جو” (jo) inflects based on the number and gender of the following noun, defaulting to nominative case in the first three examples. In the last example, its accusative form “ جي” (je) reflects the case of the intermediate noun “ پڇ” (puchh), which is accusative as it is followed by ADP, while the final “ جو” (jo) matches “ وار” (vār). Thus, the genitive ADP’s case aligns with the following noun’s case, accusative if the following noun is accusative, nominative otherwise. The number and gender also agree with the noun.

C Passive Voice Examples

Here, we continue the discussion of the passive voice started in section 4.7.

The present tense perfective aspect in Sindhi employs two distinct patterns for passive voice constructions. The first pattern uses a perfective verb form followed by “ ويو آهي” (viyo āhe, “has been”), where “ ويو” (viyo) serves as an intermediate auxiliary marking passivization, while “ آهي” (āhe) indicates present tense. For example, the active sentence “ اڪبر ڪتاب لکيو آهي” (Akbar kitāb likhyo āhe, “Akbar has written a book”) becomes “ ڪتاب اڪبر ڪان لکيو ويو آهي” (kitāb Akbar kān likhyo viyo āhe, “The book has been written by Akbar”). In this case, “ ويو” (viyo) is tagged with the UD relation *aux:pass* because it forms the passive voice, while “ آهي” (āhe) retains the *aux* relation as a tense marker. This can be seen in following graph (The obl:agent اڪبر is not shown):



The second pattern involves a perfective morphological passive form, inflected for number and gender, paired with a present tense auxiliary that solely marks tense, not passivization. Here, the auxiliary does not require the *aux:pass* relation, as the main verb's morphology already indicates the passive voice. This can be seen in " لکجيا ڪتاب آهن" (kitāb likhjiyā āhan, "Books have been written"), where " لکجيا" (likhjiyā) is a masculine plural passive form with " ڪتاب" (kitāb, "books") as *nsubj:pass*, and " آهن" (āhin) is tagged as *aux* for present tense.

In the same way past imperfective passive is formed by the passive verb form combined with past tense auxiliaries (with usual number, gender inflections) like " لکجي ها" (likhje hā, "would have written") and " لکبا هئا" (likhbā huā, "were being written"). The past perfective passive mirrors the present perfective with past tense auxiliaries.

Similarly, Sindhi future passive formations employ various patterns combining perfective or passive verb forms with future tense auxiliaries, such as ويندو" (vendo) or هوندو" (hondo), to indicate passivization.

Despite the variety of patterns, the Universal Dependencies (UD) relation patterns for future tense passive formations—including *nsubj:pass* (passive subject), *obl:agent* (oblique agent), and *aux:pass* (passive auxiliary)—remain consistent with those of the present tense passive constructions discussed earlier. The choice of UD relations depends on the auxiliary's role in marking passivity versus tense, but the syntactic structure mirrors that of the present tense.

D Use of Semgrex and Ssurgeon

As the annotation scheme changed, this project made extensive use of two tools for editing and checking the results of those edits: Semgrex and Ssurgeon (Bauer et al., 2023), and CoNLLUEditor (Heinecke, 2019). Both author groups were quite responsive in adding new features to support the needs of this particular project.

In the early stages of annotation, Future Participles were labeled as VerbForm=FutPart, an annotation that does not exist in current UD. As a first step to update the features to match the annotations described in this paper, we needed to update all existing Inf annotations to match the scheme described in section 4.6. We used figure 4 for this update.

After performing this update, we then searched for other verbs labeled Inf which may not fit the expected annotation patterns using figure 5.

Another example was an edit of a token where the intensifier was incorrectly tokenized as part of the word, which we edited with figure 6.

In order to search for nonprojective arcs, we used a Semgrex expression which searched for an arc going from a node before the current word to after the current word, then an arc going from the current word to a word on either side of that arc using a query shown in figure 7.

Searching for features, searching against features, splitting words, and searching for undirected connections are examples of some of the features added to support this project.

E Dependency Statistics

We present statistics on the dependencies present in the test set, along with the F1 scores for those dependencies from the best Stanza model.

Reln	F1	Total
acl	0.7586	40
acl:relcl	0.4286	5
advcl	0.8387	385
advmod	0.8852	405
advmod:emph	0.9329	142
amod	0.9206	472
appos	0.6667	2
aux	0.9605	436
case	0.9885	1436
cc	0.9366	244
ccomp	0.5185	48
compound	0.8696	711
conj	0.9464	327
cop	0.9564	266
dep	0.7403	62
det	0.9177	161
discourse	0.7619	11
dislocated	0.6667	4
fixed	0.9333	8
flat	0.9655	56
iobj	0.5806	21
mark	0.9119	353
nmod	0.9196	1148
nsubj	0.9136	896
nsubj:pass	0.1111	15
nummod	0.9565	88
obj	0.8565	416
obl	0.8763	790
parataxis	0.4000	3
punct	0.9465	1047
root	0.9763	887
vocative	0.6667	1
xcomp	0.6440	173

```
{word:/^.* ڻ$/;cpos:VERB;morphofeatures:{VerbForm:Inf}}=word
EditNode -node word -morphofeatures Aspect=Imp|VerbForm=Inf
```

Figure 4: An Ssurgeon pattern for updating existing infinitives to match a new feature pattern

```
{morphofeatures:{Aspect!:Imp;VerbForm:Inf}}
```

Figure 5: A Semgrex search for features which do not match the expected Infinitive guidelines

```
{word:/^ يئردنا$/}=split <=edge {}
splitWord -node split -exact ردنا -exact يئ -reln advmod:emph
    -headIndex 0 -name 0=adv,1=emph
editNode -node emph -pos PART -cpos PART -after “ ”
editNode -node adv -pos ADP -cpos ADV -remove morphofeatures
relabelNamedEdge -edge edge -reln advmod
```

Figure 6: A Ssurgeon pattern for splitting an incorrectly attached intensifier, then applying tags and dependencies

```
{} .. {}=later -- ({}=earlier <> {}=later)
    [<> ({}=attach -- {}=later) | <> ({}=attach .. {}=earlier)]
```

Figure 7: A Semgrex search for finding non-projective arcs