Vyākarana: A Colorless Green Benchmark for Syntactic Evaluation in Indic Languages

Rajaswa Patil
Cognitive Neuroscience Lab
BITS Pilani, Goa Campus
f20170334@goa.bits-pilani.ac.in

Jasleen Dhillon
Dept. of CS & IS
BITS Pilani, Goa Campus
f20170077@goa.bits-pilani.ac.in

Siddhant Mahurkar
Data Science Institute
Columbia University
sm5129@columbia.edu

Saumitra Kulkarni
Dept. of Computer Engineering
PICT, Maharashtra, India
kulkarnisaumitra98@gmail.com

Manav Malhotra
Dept. of Mathematics
BITS Pilani, Goa Campus
f20171016@goa.bits-pilani.ac.in

Veeky Baths
Cognitive Neuroscience Lab
BITS Pilani, Goa Campus
veeky@goa.bits-pilani.ac.in

Abstract

While there has been significant progress towards developing NLU resources for Indic languages, syntactic evaluation has been relatively less explored. Unlike English, Indic languages have rich morphosyntax, grammatical genders, free linear word-order, and highly inflectional morphology. In this paper, we introduce *Vyākarana*: a benchmark of Colorless Green sentences in Indic languages for syntactic evaluation of multilingual language models. The benchmark comprises four syntax-related tasks: PoS Tagging, Syntax Tree-depth Prediction, Grammatical Case Marking, and Subject-Verb Agreement. We use the datasets from the evaluation tasks to probe five multilingual language models of varying architectures for syntax in Indic languages. Due to its prevalence, we also include a code-switching setting in our experiments. Our results show that the token-level and sentence-level representations from the Indic language models (IndicBERT and MuRIL) do not capture the syntax in Indic languages as efficiently as the other highly multilingual language models. Further, our layer-wise probing experiments reveal that while mBERT, DistilmBERT, and XLM-R localize the syntax in middle layers, the Indic language models do not show such syntactic localization.

1 Introduction

The Indian subcontinent is home to more than 450 languages spanning seven language families. Multilingualism and code-switching are common phenomenon across these languages. Traditionally, the region has been at the center of many linguistic studies due to its rich linguistic diversity (Emeneau, 1956). The Indo-Aryan and the Dravidian language families are the most prominent ones in the subcontinent, with more than a billion speakers combined. Recent work in computational linguistics has focused on clubbing together the major languages from these two language families into a single group called as *Indic Languages*.[1] Even with such a huge potential user base, the progress in building language technologies for the Indic languages has been limited. Most of the Indic languages fall under the category of low-resourced and mid-resourced languages. Some recent open-sourced efforts have tried to address this by developing various toolkits (Arora, 2020), frameworks (Kunchukuttan, 2020), language models, benchmarks, and datasets (Kakwani et al., 2020).

To perform well on natural language understanding tasks, a language model should have a good understanding of the various general aspects of the structure of language, like its grammar and syntax. The structure of Indic languages is quite rich in its morphosyntax, which is partially shared across the languages. Thus, performing language modeling and downstream tasks on a multilingual dataset of Indic languages helps capture typologically generalized stimuli across the languages and implicitly addresses the issue of data-scarcity. While there have been some efforts to analyze the role of multilingual training for Indic languages, it has been limited to downstream NLU tasks and large multilingual models which are not exclusively trained for Indic languages (Jain et al., 2020). In this work, we introduce a syntactic evaluation benchmark of *Colorless Green*[2] sentences in Indic languages: *Vyākarana*[3], which provides a challenging envi-

---

[1] From a linguistic perspective, the term *Indic* is used for the Indo-Aryan language family. However, in this work, we use the term to represent all the languages used in the subcontinent (following the recent work in "Indic NLP").

[2] *Colorless green* sentences are grammatically correct, but semantically nonsensical.

[3] Vyākarana is the Sanskrit term for grammar

ronment to evaluate multilingual language models for their syntactic abilities specifically. We use the dataset to conduct layer-wise probing of five multilingual language models to inspect their understanding of the syntax in Indic languages. We include the IndicBERT (Kakwani et al., 2020) and MuRIL (Khanuja et al., 2021)[4] models in our experiments, which are trained on Indic languages and English exclusively. We probe the models for four syntax-related tasks: PoS Tagging (POS), Syntax Tree-depth Prediction (STDP), Grammatical Case Marking (GCM), and Subject-Verb Agreement (SVA). In an attempt to capture the linguistic diversity in the Indian subcontinent, we include one language each from the Indo-Aryan (Hindi) and Dravidian (Tamil) language families in the dataset. In order to incorporate the prevalent phenomena of code-switching in these languages, we also include English-based script-mixed versions of these languages in our experiments. Our layer-wise probing experiments show that the Indic language models do not show syntactic localization and do not capture the syntax in Indic languages as efficiently as the other "highly multilingual language models".[5]

2 Background

Language models are usually evaluated with information theory-based perplexity measures. While these perplexity measures might show good agreement with a language model's natural language understanding (NLU) capabilities, they do not capture language models' syntactic abilities efficiently (Tran et al., 2018). Evaluating language models' syntactic abilities is quite important in furthering the research towards developing human-like robust language models (Linzen, 2020). Recent work has focused on the targeted syntactic evaluation of language models (Linzen et al., 2016; Lau et al., 2017; Gulordava et al., 2018; Marvin and Linzen, 2018; McCoy et al., 2019; Futrell et al., 2019), which takes inspiration from various psycholinguistic generalizations found in humans and assesses the role of syntax in the models' ability to perform various NLU tasks. While there has been significant progress towards building NLU evaluation benchmarks (Wang et al., 2018, 2019), the work in developing syntactic evaluation benchmarks has been quite recent and limited (Gauthier et al., 2020; Hu et al., 2020a; Mueller et al., 2020).

Probing is an alternative paradigm that can be used for syntactic evaluation of language models. Probing deals with quantifying the amount of linguistic information encoded in the pre-trained representations of language models (Adi et al., 2017; Conneau et al., 2018). While this does not provide a very detailed analysis of models' syntactic behavior, it can be used to compare the amount of syntactic information captured by the models, as well as the inner dynamics of how and where this information is encoded in the models (Hewitt and Manning, 2019; Jawahar et al., 2019; Lin et al., 2019; Liu et al., 2019; Tenney et al., 2019b; Rogers et al., 2020). Probing methods are also relatively compute-efficient, as they do not involve training or fine-tuning the language models.

With the recent development in multilingual language modeling, numerous evaluation studies have been performed to test the models' multilingual and cross-lingual abilities (Libovický et al., 2019; Rönnqvist et al., 2019; Pires et al., 2019; Wu and Dredze, 2019; Artetxe et al., 2020). Similar to monolingual language modeling paradigms, there has been significant work towards building multilingual evaluation benchmarks (Hu et al., 2020b; Kakwani et al., 2020; Liang et al., 2020). However, these benchmarks do not cover syntactic evaluation efficiently, with POS tagging being the sole syntax-related task. Some recent studies have tried to address this by building multilingual and cross-lingual syntactic evaluation suites with subject-verb agreement tasks (Gulordava et al., 2018; Mueller et al., 2020).

Alongside the developments in multilingual NLP, there has been some recent progress towards advancing Indic NLP (Arora, 2020; Jain et al., 2020; Kakwani et al., 2020; Kunchukuttan, 2020). While there are numerous datasets available to benchmark and compare multilingual language models on NLU tasks in Indic languages, there is no such resource available for syntactic evaluation in Indic languages.[6] Most recent works (Jain et al., 2020; Kakwani et al., 2020; Khanuja et al., 2021) rely on POS tagging with Universal Dependencies treebanks as the sole task to compare the syntactic abilities of Indic language models.

---

[4]Collectively called as "Indic language models" here on
[5]mBERT, DistilmBERT, and XLM-R
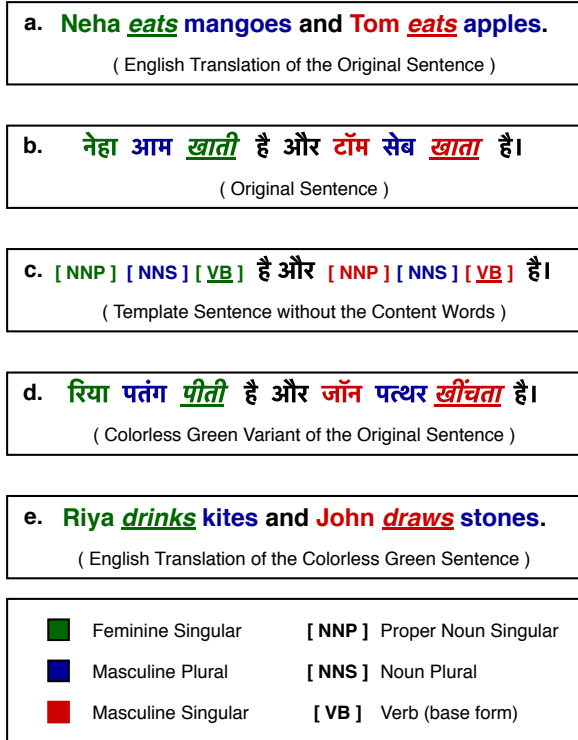
[6]To the best of our knowledge

Figure 1: (a,b) - Comparing the linear word-order in English and Hindi; (b,c,d,e) - Constructing Colorless Green sentences in Hindi.

3 Vyākarana: A Colorless Green Benchmark[7]

Syntax in Indic languages has specific characteristics which make it more challenging to model. Unlike English, most of the Indic languages follow a free linear word-order, with the default order being SOV (subject-object-verb). Hence, the linear distance (in terms of the number of intervening words) between the verb and its subject is usually larger in Indic languages as compared to English (Figure 1a; Figure 1b). Indic languages like Hindi and Tamil also have grammatical genders and grammatical number, where the words are morphologically inflected in agreement with the gender and count of their corresponding head nouns (Figure 1b).

Usually, the syntactic evaluation of language models is conducted with syntax-related probing tasks (Jawahar et al., 2019), or with targeted syntactic evaluation through controlled psycholinguistic experiments (Gauthier et al., 2020). Performing such syntactic evaluations under a colorless green setting ensures that the models do not leverage any semantic or lexical cues and biases from the text

to process its syntactic structure (Gulordava et al., 2018; Goldberg, 2019). Most of the evaluation tasks from both of these categories can be designed with a syntactic dependencies treebank. Hence, we construct a colorless green treebank for Indic languages, which can be used to perform various syntactic evaluation tasks under a colorless green setting.

3.1 CG-HDTB: A Colorless Green Treebank for Hindi[8]

We use the Hindi Universal Dependency Treebank (HDTB) (Bhat et al., 2008; Palmer et al., 2009) to construct a new Colorless Green treebank for Hindi: *CG-HDTB*.[9] We follow the method proposed by Gulordava et al. (2018) to generate colorless green sentences. Given a sentence from the original HDTB treebank, we first convert it to a template colorless green sentence (Figure 1c) by removing all the content words (while retaining all the function words) from it. Each of the resulting gaps created by the removal of content words in the template sentence is then populated with a content word from another sentence in the treebank, where the grammatical information of the original and replacing content word is the same (Figure 1d). The resulting colorless green sentence still retains the same grammatical information and syntactic structure as the original sentence, whereas the random substitution of the content words ensures that it is rendered nonsensical (Figure 1a; Figure 1e).

PoS Tag	Gen	Num	Case	Per
Noun	✓	✓	✓	
Verb	✓	✓		✓
Adjective	✓	✓	✓	
Adverb	✓	✓	✓	

Table 1: The grammatical information taken into account for various PoS categories of content words (Vikram, 2013).

The grammatical information taken into consideration while substituting the content words includes the Grammatical Gender **(Gen)** (masculine/feminine), Grammatical Number **(Num)** (singular/plural), Grammatical Case **(Case)** (Sec-

---

[7]Example sentences from the benchmark dataset can be found in Appendix A

[8]In this section we use Hindi as a demonstrative language, where a similar approach is used for Tamil language.

[9]Similarly, we use the Tamil Universal Dependency Treebank (TTB) (Ramasamy and Žabokrtský, 2012) to construct a new Colorless Green treebank for Tamil: *CG-TTB*.

Data Split	HDTB			CG-HDTB		
	Sentences	Masculine	Feminine	Sentences	Masculine	Feminine
Train	13304	104389 (70.37%)	43951 (29.63%)	6736	39178 (51.72%)	36578 (48.28%)
Dev	1659	13116 (70.47%)	5496 (29.53%)	6628	38638 (51.90%)	35810 (48.10%)
Test	1684	13253 (69.98%)	5686 (30.02%)	53124	308034 (51.91%)	285322 (48.09%)

Table 2: The number of sentences and the statistics for token-level grammatical gender feature in the HDTB and CG-HDTB treebank (the statistics are same for CG-HDTB and csCG-HDTB treebanks).

tion 3.2), and Person (**Per**) (first/second/third). This makes sure that the replacing content word is consistent with syntax and the morphological inflections of the original content word, keeping the morphosyntax of the sentence intact. The grammatical information required for every PoS category is different, as shown in Table 1. The HDTB treebank has an imbalanced count of Grammatical Gender features. Following an uncontrolled and random substitution of content words can result in a gender-imbalanced dataset of colorless green sentences. We eliminate this imbalance by controlling the grammatical gender information in the colorless green sentences. For every sentence in the HDTB treebank, we generate four colorless green sentences, where the gender encoding of the sentences is:

1. Exactly **same** as that of the original sentence.
2. Exactly **opposite** as that of the original sentence.
3. Entirely **masculine**.
4. Entirely **feminine**.

Apart from the PoS categories of content words mentioned in Table 1, we also adjust the gender-inflected Adposition tokens in the colorless green sentences in order to maintain the morphosyntax of the sentence. The statistics for the Grammatical Gender feature in the gender-imbalanced HDTB treebank and the gender-balanced CG-HDTB treebank are shown in Table 2. Further, we swap the train and test set of the HDTB treebank while generating colorless green sentences (Table 2). This allows the test set of CG-HDTB to be significantly bigger than the training set, providing a challenging test setting for the models' syntactic generalization capabilities.

Unlike Hindi, there is no publicly available treebank for Hindi-English code-switched setting with all the features that are required to obtain the colorless green sentences. Hence, we artificially create a code-switched treebank for Hindi-English: **csCG-HDTB**. The csCG-HDTB treebank is a parallel code-switched version of the CG-HDTB treebank. Following Khanuja et al. (2021)'s approach, we use the Indic-Trans library[10] to transliterate the tokens (Hindi-Devanagari script → English-Latin script) from each sentence in the CG-HDTB treebank.[11] While this only helps in incorporating the script-mixing aspect of code-switching, the approach can be scaled easily across different datasets and languages, eliminating the need of explicit annotations of morphosyntactic features on code-switched data.

3.2 Evaluation Tasks

The benchmark comprises four syntactic evaluation tasks. We borrow certain design principles for the benchmark from the XTREME benchmark Hu et al. (2020b):

1. **Task Difficulty:** The colorless-green setting, and a sufficiently large test-set ensure a certain level of difficulty across all the tasks. Further, some tasks require capturing long-range relationships and morphological inflections, which make them more challenging.
2. **Task Diversity:** We include both the token-level and the sentence-level evaluation tasks in the benchmark. This ensures that the benchmark evaluates representations from both the granularities.
3. **Data and Training Efficiency:** Under the colorless-green setting, each task has a large test-train ratio in terms of data samples (Table 3). Hence, the datasets for the evaluation tasks are quite challenging in terms of data and training efficiency.

We use the HDTB, CG-HDTB, and csCG-HDTB treebanks to construct datasets for all the evaluation tasks.[12] The statistics for the con-

---

[10]https://github.com/libindic/indic-trans

[11]Similarly, we create a code-switched treebank for Tamil-English: **csCG-TTB**.

[12]Similarly, we use the CG-TTB and csCG-TTB treebanks to conduct the experiments with the Tamil language (Appendix B).

Data Split	HDTB				CG-HDTB			
	POS	STDP	GCM	SVA	POS	STDP	GCM	SVA
Train	281057	13304	151275	17034	141720	6736	76836	8536
Dev	35430	1659	19209	2127	140844	6628	76108	8508
Test	35217	1684	19027	2134	1123964	53124	605068	68136

Table 3: The number of tokens (for POS and GCM tasks) and the number of sentences (for STDP and SVA tasks) in the HDTB and CG-HDTB treebank (the statistics are same for CG-HDTB and csCG-HDTB treebanks).

structed datasets are given in Table 3.

PoS Tagging (POS): While PoS tagging is a very weak construct for a syntactic evaluation task, it is a preliminary step in syntactic processing. It is also the only existing syntax-related evaluation task used to compare currently available Indic language models. Hence, we include PoS tagging as the first evaluation task in the benchmark. The task is designed under a token-level multi-class single-label classification setting. We use the UPOS tags from the treebanks as the ground-truth labels for this task.

Syntax Tree-depth Prediction (STDP): Following the work done by Conneau et al. (2018) and Jawahar et al. (2019), we use the dependency trees from the treebanks to perform syntax tree-depth prediction task. Successful prediction of the depth of a dependency tree depicts the model's ability to get a surface-level estimate of a given sentence's syntactic structure. The task is designed under a sentence-level multi-class single-label classification setting.

Grammatical Case Marking (GCM): Given the nature of morphosyntax in Indic languages, case marking is an essential syntactic evaluation task. Unlike languages like English, a fixed linear word-order cannot be used to perform case marking in Indic languages. Instead, one must rely on morphological inflections and Adposition tokens in the sentence to assign appropriate grammatical cases. Moreover, a single token can be marked with a combination of multiple cases. In the HDTB treebank, the grammatical case feature has the following seven unique values: **accusative**, **nominative**, **accusative-inessive**, **dative-accusative**, **ergative-accusative**, **genitive-accusative**, and **instrumental-accusative**. Hence, this task is designed under a token-level multi-class single-label classification setting.

Subject-Verb Agreement (SVA): We include subject-verb agreement as an evaluation task to study the long-range syntactic dependencies in Indic languages. Unlike English, the subject-verb agreement in Indic languages is dependent on the grammatical number as well as the grammatical gender of the tokens. Following the work done by Linzen et al. (2016), we include all the tokens preceding (and excluding) the target verb in the sentence. Given such a sentence, the task is to predict the target verb's grammatical count and grammatical gender in agreement with its subject (head noun). Given that most of the Indic languages follow a free linear word-order and the default order is SOV (subject-object-verb), the number of intervening nouns is significantly higher than in English. Moreover, there is a high probability of the object acting as an attractor noun (Figure 1). Hence, the SVA task is significantly more challenging in Indic languages. The task is designed under a sentence-level multi-class single-label classification setting, given the four possible ground truth labels: **masculine-singular**, **masculine-plural**, **feminine-singular**, and **feminine-plural**.

4 Experiments[13]

4.1 Experimental Setup

We use the datasets constructed for the evaluation tasks to perform layer-wise probing[14] of token-level and sentence-level representations from the five transformer-based multilingual language models: mBERT (Devlin et al., 2019), XLM-R (Conneau et al., 2020), DistilmBERT (Sanh et al., 2019), IndicBERT (Kakwani et al., 2020), and MuRIL (Khanuja et al., 2021). We use a single linear-layer (initialized with the same weights across all the ex-

---

[13]In this section we discuss the experiments with the Hindi language data. Corresponding details about the experiments with the Tamil language data can be found in Appendix B.

[14]We do not fine-tune the models due to computational limitations. Instead, we report the probing metrics of the last layer and the best-performing layer of every model (Table 4).

Treebank	Task	Last Layer					Best Layer				
		mBERT	XLM-R	DistilmBERT	IndicBERT	MuRIL	mBERT	XLM-R	DistilmBERT	IndicBERT	MuRIL
HDTB	POS	0.9332	**0.9567**	0.8860	0.7827	0.7290	0.9409 (8)	**0.9615 (8)**	0.8955 (5)	0.8105 (2)	0.7728 (1)
	STDP	**0.3886**	0.1848	0.3815	0.3825	0.2588	0.4138 (3)	0.3657 (1)	**0.4247 (3)**	0.3995 (3)	0.3969 (8)
	GCM	0.7501	**0.7733**	0.7345	0.6509	0.6350	0.7775 (8)	**0.8006 (8)**	0.7495 (4)	0.6847 (3)	0.6522 (1)
	SVA	**0.7083**	0.6208	0.7015	0.5122	0.5178	**0.7083 (12)**	0.6208 (12)	0.7015 (6)	0.5125 (3)	0.5718 (12)
	Average	**0.6951**	0.6339	0.6759	0.5821	0.5352	**0.7101**	0.6872	0.6928	0.6018	0.5984
CG-HDTB	POS	0.8894	**0.9143**	0.8448	0.7418	0.6807	0.8932 (11)	**0.9232 (8)**	0.8543 (5)	0.7642 (6)	0.7030 (2)
	STDP	0.3455	0.2138	0.3254	**0.3566**	0.2318	0.3730 (6)	0.3232 (3)	**0.3881 (5)**	0.3756 (4)	0.3570 (10)
	GCM	0.6886	**0.6967**	0.6733	0.6216	0.5463	0.7050 (7)	**0.7234 (8)**	0.6968 (4)	0.6479 (2)	0.5878 (3)
	SVA	0.6003	0.5935	**0.6040**	0.3990	0.5568	**0.6140 (8)**	0.5935 (12)	0.6040 (6)	0.4451 (2)	0.5568 (12)
	Average	**0.6310**	0.6046	0.6119	0.5298	0.5039	**0.6463**	0.6408	0.6358	0.5582	0.5511
csCG-HDTB	POS	0.7543	**0.7881**	0.7422	0.7303	0.6752	0.7744 (7)	**0.8079 (10)**	0.7596 (5)	0.7608 (2)	0.7011 (1)
	STDP	0.3104	0.3429	0.3531	**0.3805**	0.2457	0.3879 (7)	0.3584 (3)	0.3843 (5)	**0.3909 (3)**	0.3676 (7)
	GCM	0.6415	0.6456	**0.6533**	0.6306	0.5653	0.6604 (8)	**0.6683 (8)**	0.6549 (5)	0.6535 (8)	0.5859 (3)
	SVA	0.5334	**0.5629**	0.5255	0.5389	0.5262	0.5650 (1)	0.5629 (12)	**0.5752 (3)**	0.5389 (12)	0.5647 (11)
	Average	0.5599	**0.5849**	0.5685	0.5701	0.5031	0.5969	**0.5994**	0.5935	0.5860	0.5548

Table 4: The weighted-F1 scores for the last layer and the best layer (layer-number mentioned in parenthesis) for the layer-wise probing experiments with the Hindi language data. (Corresponding results for the Tamil language data can be found in Appendix B.1)

periments) as the probing classifier for all the tasks and models. Since POS and GCM are token-level tasks, we use the first-subword token-embeddings as the input to the probing classifier. Whereas we use the special sentence-token embeddings for the sentence-level STDP and SVA tasks. We use Hugging Face's *transformers* library (Wolf et al., 2020) to access the pre-trained instances of these language models. The probing classifier is trained using the PyTorch library (Paszke et al., 2019). Given that all the evaluation tasks are designed under a multi-class single-label classification setting, we monitor the weighted-F1 scores to evaluate the models (Table 4).

4.2 Probing Results

Overall, due to the lack of semantic and lexical cues, and a larger test-set, all the models find the CG-HDTB data more challenging than the HDTB data (Table 4). The performance across all the models and tasks further deteriorates with the code-switched csCG-HDTB data. Given the considerably larger scale (model and pre-training data) of the XLM-R model, for the token-level tasks (POS and GCM), it significantly outperforms the other models for all the three treebanks. Whereas, for the sentence-level tasks (STDP and SVA), mBERT and DistilmBERT perform better than the rest of the models for HDTB and CG-HDTB data. This might be explained by the sentence-level pre-training tasks (Next Sentence Prediction) used by mBERT and DistilmBERT. Overall, for Hindi data, mBERT shows the best average weighted-F1 scores across the four tasks, and the Indic models perform the

worst. On the other hand, XLM-R significantly outperforms the rest of the models under a code-switched Hindi-English setting (Table 4).

Even though the IndicBERT and MuRIL models outperform mBERT and XLM-R on many downstream NLU tasks (Kakwani et al., 2020; Khanuja et al., 2021), they consistently fail to catch-up in syntactic evaluation. Even DistilmBERT, a relatively smaller model, outperforms IndicBERT and MuRIL across all the tasks. There might be multiple plausible reasons behind this finding. Both IndicBERT, and MuRIL perform masked word/token language modeling and do not have a sentence-level pre-training task. Hence, while they are outperformed by mBERT and DistilmBERT, they outperform XLM-R on most of the sentence-level tasks with Hindi data. Moreover, they are significantly smaller in architecture size (IndicBERT) and dataset size as compared to the other models. mBERT, XLM-R, and DistilmBERT are highly multilingual language models, pre-trained on more than 100 languages. This might provide them with the linguistic and typological generalization required for modeling morphosyntax more efficiently than Indic models, which are only trained on a handful of Indic languages and English. Even though the MuRIL model is pre-trained with artificially generated parallel translated and transliterated sentences in Indic languages and their English counterparts, it is outperformed by XLM-R on the csCG-HDTB Hindi-English code-switched data. The XLM-R model has a small amount of natural code-switched data in its pre-training corpus, which might be the reason behind its dominance

under the code-switching setting.

4.3 Localization of Syntactic Knowledge

It has been previously shown that linguistic knowledge is localized in the monolingual BERT (Rogers et al., 2020). While surface-level linguistic information like linear-word order and sentence-length is captured in the lower layers of the model (Jawahar et al., 2019; Lin et al., 2019), the syntactic knowledge is found in the middle-layers of the model (Hewitt and Manning, 2019; Jawahar et al., 2019). While basic syntactic operations are encoded in relatively lower-layers (Jawahar et al., 2019; Tenney et al., 2019a), more complex tasks with long-range syntactic dependencies are performed best by relatively higher layers (Goldberg, 2019). Rogers et al. (2020) explain this localization of syntactic knowledge using Liu et al. (2019)'s finding that the middle layers of BERT are most generalizing across tasks and the deeper layers are more task-specific. We investigate the localization of syntactic knowledge in multilingual language models by plotting the layer-wise performance across the various evaluation tasks with CG-HDTB data (Figure 2).

For token-level tasks (POS and GCM) (Figure 2a 2c), we find a pattern similar to that of monolingual BERT in mBERT, DistilmBERT, and XLM-R, where the weighted-F1 scores peak at the middle layers. The same, however, is not true for IndicBERT and MuRIL, which show declining weighted-F1 scores with increasing depth in the model. This suggests that with increasing depth, the layers become more task-specific. This is an unusual behavior for the Indic models, where investigating the cause for this should be an interesting research direction. The XLM-R model consistently outperforms the rest of the models across all the layers. Whereas the Indic language models consistently fall behind the other highly multilingual models. Between the Indic models, IndicBERT significantly outperforms MuRIL across all the layers.

For sentence-level tasks (STDP and SVA) (Figure 2b 2d), XLM-R, IndicBERT and MuRIL show no clear localization pattern[15]. mBERT and DistilmBERT show a consistent performance across all the layers, outperforming the rest of the models for both tasks. Similar to Jawahar et al. (2019)'s findings of the STDP task with monolingual BERT, they peak at the middle layers. However, the difference between the layers' performance is not very

---

[15](with the exception of IndicBERT on task STDP)

significant (Figure 2b). IndicBERT performs particularly well on the STDP task, showing comparable metrics with mBERT and DistilmBERT. While MuRIL starts lowest for STDP task, it eventually manages to outperform XLM-R, but still consistently lags behind IndicBERT (Figure 2b). Whereas, for SVA task, MuRIL performs better than IndicBERT for most of the layers (Figure 2d). XLM-R model shows a dip in performance in the middle layers for both the tasks, which is exactly opposite to that of the token-level tasks and previous findings with monolingual BERT.

5 Conclusion

This work presents a gender-balanced benchmark evaluation of Colorless Green sentences in Indic languages for syntactic testing of multilingual language models. By doing so, we aim to address the existing research gap in syntactic testing in Indic NLP. We introduce four new treebanks: CG-HDTB, csCG-HDTB, CG-TTB, and csCG-TTB for this purpose. We use the datasets from the treebanks to perform four syntax-related benchmarking evaluation tasks. In our experiments, we perform layer-wise probing of token-level and sentence-level representations from five different multilingual language models. Our experiments reveal that performing syntactic evaluation under a colorless green setting ensures that the semantic and lexical cues do not add evaluation artifacts in the form of higher-than-actual performance metrics. Further, the experiments also reveal that the multilingual language models suffer significantly under a code-switched setting, which is an important aspect in Indic languages. Overall, our experiments show that the currently available Indic language models do not capture syntax as efficiently as the other highly multilingual language models. Indic models seem to outperform other multilingual language models on various NLU tasks in Indic languages even with a certain lack of syntactic knowledge. This might suggest that the Indic language models do not rely significantly on syntax while making inferences on such NLU tasks. We also observe that training on large artificial corpora of transliterated texts do not help Indic language models in capturing syntax under a code-switched setting.

Further, we find that unlike the other highly multilingual language models, the currently available Indic language models do not show any syntactic localization in the middle layers. This is a unique
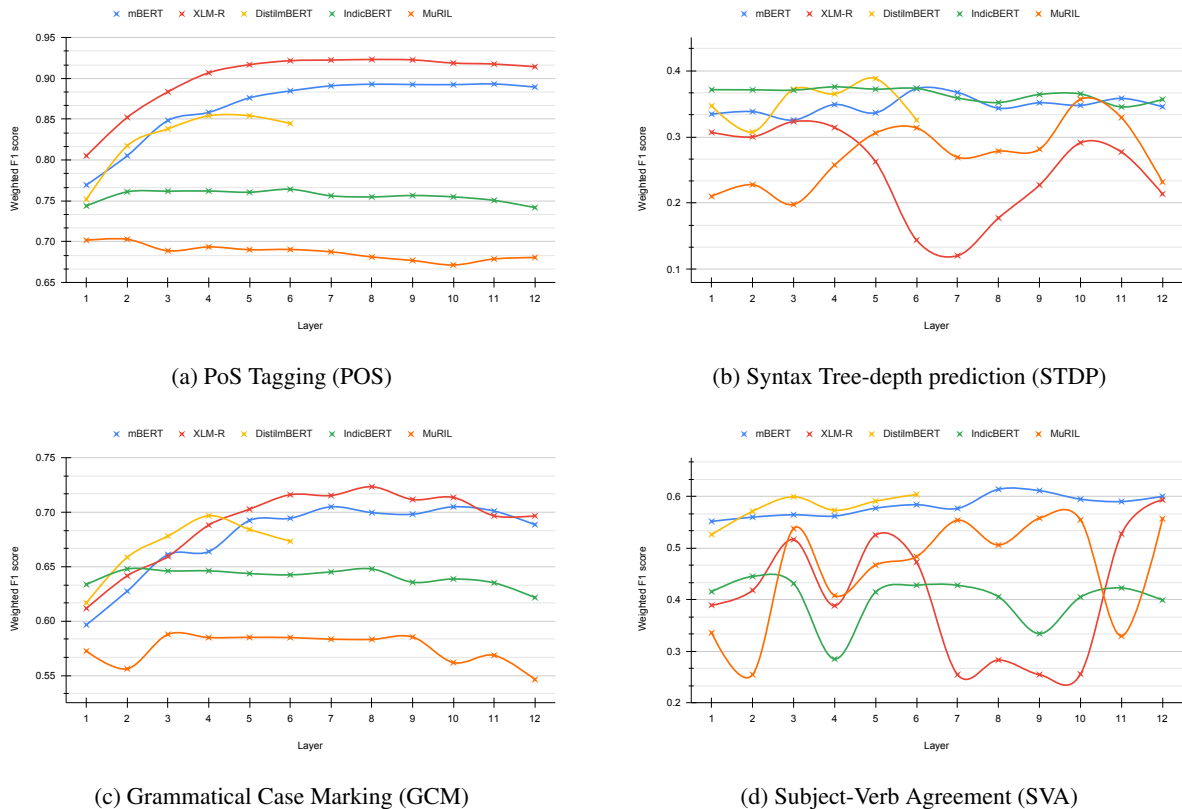
(a) PoS Tagging (POS)

(b) Syntax Tree-depth prediction (STDP)

(c) Grammatical Case Marking (GCM)

(d) Subject-Verb Agreement (SVA)

Figure 2: Weighted F1 scores for the layer-wise probing experiments with the CG-HDTB test set for the mBERT, DistilmBERT, XLM-R, IndicBERT, and MuRIL models.

behavior to the multilingual Indic language models, the cause and effects for which can be investigated in another independent study. While the current work only covers the Hindi and Tamil languages, it lays down the framework for performing syntactic evaluation in other Indic languages.

6 Development and Accessibility

As a continuation of the current work, we aim to cover more Indic languages and their code-switched counterparts in the benchmark. We aim to cover typologically different languages from both the Indo-Aryan, and the Dravidian language families, with a special focus on low-resourced Indic languages. This can be an extremely challenging task. Developing datasets for syntactic evaluation requires a certain level of linguistic expertise, which is usually found in the trained and native speakers of that particular language. While the methods used in this work relax these requirements upto some extent, they are dependent on existing syntactic dependency annotations. Such fine-grained annotations are not readily available for a majority of the Indic languages. Hence, we plan to open-source the benchmark datasets. We

aim to continue developing the benchmark with open-source contributions from the trained and native speakers of various Indic languages.

The treebank datasets used for this benchmark are publicly available for benchmarking and development purposes on GitHub and the Hugging Face Datasets platform (Lhoest et al., 2021).[16]

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

[16]https://github.com/rajaswa/
indic-syntax-evaluation

160

Gaurav Arora. 2020. iNLTK: Natural language toolkit for indic languages. In *Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)*, pages 66–71, Online. Association for Computational Linguistics.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

Riyaz Ahmad Bhat, Rajesh Bhatt, Annahita Farudi, Prescott Klassen, Bhuvana Narasimhan, Martha Palmer, Owen Rambow, Dipti Misra Sharma, Ashwini Vaidya, Sri Ramagurumurthy Vishnu, et al. 2008. The hindi/urdu treebank project. In *Handbook of Linguistic Annotation*. Springer Press.

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. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

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.

M. B. Emeneau. 1956. India as a linguistic area. *Language*, 32(1):3–16.

Richard Futrell, Ethan Wilcox, Takashi Morita, Peng Qian, Miguel Ballesteros, and Roger Levy. 2019. Neural language models as psycholinguistic subjects: Representations of syntactic state. 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 32–42, Minneapolis, Minnesota. Association for Computational Linguistics.

Jon Gauthier, Jennifer Hu, Ethan Wilcox, Peng Qian, and Roger Levy. 2020. SyntaxGym: An online platform for targeted evaluation of language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 70–76, Online. Association for Computational Linguistics.

Yoav Goldberg. 2019. Assessing bert's syntactic abilities. *CoRR*, abs/1901.05287.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205, New Orleans, Louisiana. Association for Computational Linguistics.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. 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 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger Levy. 2020a. A systematic assessment of syntactic generalization in neural language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1725–1744, Online. Association for Computational Linguistics.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020b. XTREME: A massively multilingual multitask benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.

Kushal Jain, Adwait Deshpande, Kumar Shridhar, Felix Laumann, and Ayushman Dash. 2020. Indictransformers: An analysis of transformer language models for indian languages.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961, Online. Association for Computational Linguistics.

161

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.

Anoop Kunchukuttan. 2020. The IndicNLP Library. https://github.com/anoopkunchukuttan/indic_nlp_library/blob/master/docs/indicnlp.pdf.

Jey Han Lau, Alexander Clark, and Shalom Lappin. 2017. Grammaticality, acceptability, and probability: A probabilistic view of linguistic knowledge. *Cognitive Science*, 41(5):1202–1241.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander M. Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing.

Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, Xiaodong Fan, Ruofei Zhang, Rahul Agrawal, Edward Cui, Sining Wei, Taroon Bharti, Ying Qiao, Jiun-Hung Chen, Winnie Wu, Shuguang Liu, Fan Yang, Daniel Campos, Rangan Majumder, and Ming Zhou. 2020. XGLUE: A new benchmark datasetfor cross-lingual pre-training, understanding and generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6008–6018, Online. Association for Computational Linguistics.

Jindrich Libovický, Rudolf Rosa, and Alexander Fraser. 2019. How language-neutral is multilingual bert? *CoRR*, abs/1911.03310.

Yongjie Lin, Yi Chern Tan, and Robert Frank. 2019. Open sesame: Getting inside BERT's linguistic knowledge. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 241–253, Florence, Italy. Association for Computational Linguistics.

Tal Linzen. 2020. How can we accelerate progress towards human-like linguistic generalization? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5210–5217, Online. Association for Computational Linguistics.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. Linguistic knowledge and transferability of contextual representations. 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 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Aaron Mueller, Garrett Nicolai, Panayiota Petrou-Zeniou, Natalia Talmina, and Tal Linzen. 2020. Cross-linguistic syntactic evaluation of word prediction models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5523–5539, Online. Association for Computational Linguistics.

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.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Loganathan Ramasamy and Zdeněk Žabokrtský. 2012. Prague dependency style treebank for Tamil. In *Proceedings of the Eighth International Conference*

on *Language Resources and Evaluation (LREC'12)*, pages 1888–1894, Istanbul, Turkey. European Language Resources Association (ELRA).

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Samuel Rönnqvist, Jenna Kanerva, Tapio Salakoski, and Filip Ginter. 2019. Is multilingual BERT fluent in language generation? In *Proceedings of the First NLPL Workshop on Deep Learning for Natural Language Processing*, pages 29–36, Turku, Finland. Linköping University Electronic Press.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019a. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. 2019b. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.

Ke Tran, Arianna Bisazza, and Christof Monz. 2018. The importance of being recurrent for modeling hierarchical structure. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4731–4736, Brussels, Belgium. Association for Computational Linguistics.

Shweta Vikram. 2013. Morphology: Indian languages and european languages. *International Journal of Scientific and Research Publications*, 3(6):1–5.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *CoRR*, abs/1905.00537.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 833–844, Hong Kong, China. Association for Computational Linguistics.

A Dataset Examples

	Original:	भैरव का स्थान उत्तराखंड में क्षेत्रपाल अथवा भूमिदेव के रूप में महत्वपूर्ण है ।	Transliteration:
CGI-A:	जम्मू के ग्राहक कांड में अफगानिस्तान अथवा नोएडा के रूप में महत्वपूर्ण है ।	→	jammu ke graahak cond main afghanistan athwa noida ke rup main mahatvapurn he .
CGI-B:	भारती की ड्यूटी पुलिस में अंजलि अथवा प्रदर्शनी के रूप में महत्वपूर्ण है ।	→	bharti kii duty police main anjali athwa pradarshnee ke rup main mahatvapurn he .
CGI-C:	बदरीनाथ के दृष्या अक्तूबर में प्लान अथवा सीआरपीसी के रूप में महत्वपूर्ण है ।	→	badrinath ke drushtya aktubar main plan athwa siarpisi ke rup main mahatvapurn he .
CGI-D:	क्रांति की क़दर पार्वती में सोसाइटी अथवा अंग्रेजी के रूप में महत्वपूर्ण है ।	→	kranti kii qadar parvati main socity athwa angreji ke rup main mahatvapurn he .

Figure 3: Example sentences from the CG-HDTB (left) and csCG-HDTB (right) treebank corresponding to an original sentence from the HDTB (top-left) treebank.

	Original:	பிகாரிலிருந்து ஏராளமான இளைஞர்கள் வேலை தேடி வெளி மாநிலங்களுக்கு குடிபெயர்ந்து வருகின்றனர் .	Transliteration:
CGI-A:	பிகாரிலிருந்து ஏராளமான ஆகியோர் காலம் தேடி வெளி முயற்சிகளுக்கு குடிபெயர்ந்து வருகின்றனர் .	→	bigarilrundhu bihar ilrundhu erauman akhiyor kalam theti weli mushcigaluku gudipeyarndhu varukinsener .
CGI-B:	பிகாரிலிருந்து ஏராளமான போலீஸார் வழக்கு தேடி வெளி முயற்சிகளுக்கு குடிபெயர்ந்து வருகின்றனர் .	→	bigarilrundhu bihar ilrundhu erauman polisar vanku theti weli mushcigaluku gudipeyarndhu varukinsener .
CGI-C:	பிகாரிலிருந்து ஏராளமான மனைவியர் வெளியுறவுத்துறை தேடி வெளி ஆண்டுகளுக்குப் குடிபெயர்ந்து வருகின்றனர் .	→	bigarilrundhu bihar ilrundhu erauman manavier veleuuddurai theti weli andugalukqup gudipeyarndhu varukinsener .
CGI-D:	பிகாரிலிருந்து ஏராளமான உறுப்பினர்கள் ரத்து தேடி வெளி முயற்சிகளுக்கு குடிபெயர்ந்து வருகின்றனர் .	→	bigarilrundhu bihar ilrundhu erauman ushupinars rattu theti weli mushcigaluku gudipeyarndhu varukinsener .

Figure 4: Example sentences from the CG-TTB (left) and csCG-TTB (right) treebank corresponding to an original sentence from the TTB (top-left) treebank.

Treebank	Task	Last Layer					Best Layer				
		mBERT	**XLM-R**	**DistilmBERT**	**IndicBERT**	**MuRIL**	**mBERT**	**XLM-R**	**DistilmBERT**	**IndicBERT**	**MuRIL**
CG-TTB	POS	**0.7444**	0.7336	0.6431	0.5874	0.4539	0.7719 (9)	**0.7946 (7)**	0.6809 (4)	0.6327 (6)	0.4741 (5)
	STDP	0.0947	0.0716	**0.2315**	0.0716	0.0898	0.2209 (10)	0.0898 (6)	**0.2819 (3)**	0.0716 (1)	0.2051 (11)
	GCM	**0.7319**	0.6800	0.6336	0.5864	0.5878	0.7966 (7)	**0.8187 (6)**	0.6765 (4)	0.5864 (12)	0.6040 (1)
	Average	**0.5237**	0.4951	0.5027	0.4151	0.3772	**0.5965**	0.5677	0.5464	0.4302	0.4277
csCG-TTB	POS	**0.5740**	0.5416	0.5531	0.5024	0.4526	**0.6012 (6)**	0.5947 (6)	0.5759 (5)	0.5516 (8)	0.4817 (4)
	STDP	0.1029	0.1184	**0.1476**	0.0716	0.0716	0.2431 (11)	0.1085 (1)	**0.2556 (4)**	0.0716 (12)	0.1467 (7)
	GCM	0.5931	0.5832	0.5627	0.5605	0.5717	0.6300 (11)	0.6065 (8)	0.5811 (2)	0.5852 (4)	0.5875 (7)
	Average	**0.4233**	0.4144	0.4211	0.3782	0.3653	**0.4914**	0.4366	0.4709	0.4028	0.4053

Table 5: The weighted-F1 scores for the last layer and the best layer (layer-number mentioned in parenthesis) for the layer-wise probing experiments with the Tamil language data.

B Experiments with Tamil Language

Due to limitations from the original TTB Tamil Universal Dependencies treebank, we do not cover certain grammatical-gender-based methods and experiments with Tamil language in the current work.[17] Here, we report all the other results for the CG-TTB and csCG-TTB treebank datasets.

B.1 Probing Results

While comparing the performance between the models, all the observations and inferences discussed in Section 4.2 also hold true for the Tamil language data as seen in Table 5. Overall, we observe a relatively lower performance in the CG-TTB and csCG-TTB datasets (Table 5) as compared to CG-HDTB and csCG-HDTB datasets respectively (Table 4).

B.2 Localization of Syntactic Knowledge

All the observations and inferences discussed in Section 4.3 also hold true for the Tamil language data as seen in Figure 5, Figure 6, and Figure 7. Due to a relatively lower performance on Tamil language data, all the figures are shifted downwards with almost similar trends and structures.
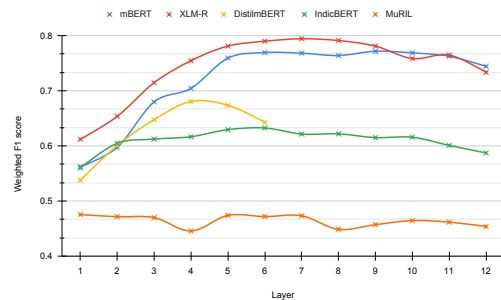

Figure 5: Weighted F1 scores for the layer-wise probing experiments with the CG-TTB test set for the POS task.
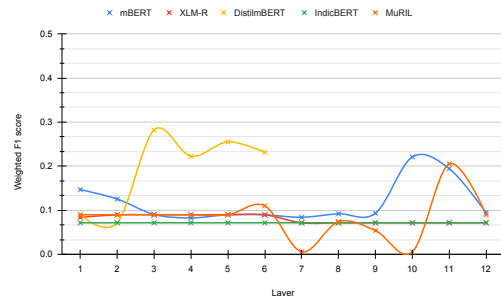

Figure 6: Weighted F1 scores for the layer-wise probing experiments with the CG-TTB test set for the STDP task.
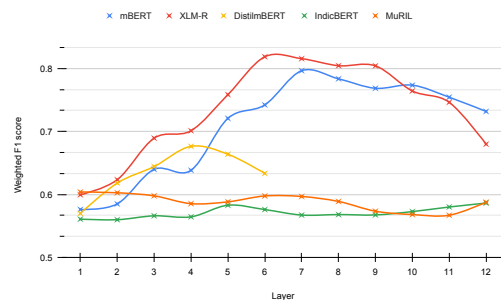

Figure 7: Weighted F1 scores for the layer-wise probing experiments with the CG-TTB test set for the GCM task.

[17]This includes the controlled generation of colorless green sentences with gender-balancing, and the Subject-Verb Agreement (SVA) task.