Systematic Investigation of Strategies Tailored for Low-Resource Settings for Sanskrit Dependency Parsing

JIVNESH SANDHAN, IIT Kanpur, India
LAXMIDHAR BEHERA, IIT Kanpur, India
PAWAN GOYAL, IIT Kharagpur, India

Existing state of the art approaches for Sanskrit Dependency Parsing (SDP), are hybrid in nature, and rely on a lexicon-driven shallow parser for linguistically motivated feature engineering. However, these methods fail to handle out of vocabulary (OOV) words, which limits their applicability in realistic scenarios. On the other hand, purely data-driven approaches do not match the performance of hybrid approaches due to the labelled data sparsity. Thus, in this work, we investigate the following question: How far can we push a purely data-driven approach using recently proposed strategies for low-resource settings? We experiment with five strategies, namely, data augmentation, sequential transfer learning, cross-lingual/mono-lingual pretraining, multi-task learning and self-training. Our proposed ensembled system outperforms the purely data-driven state of the art system by 2.8/3.9 points (Unlabelled Attachment Score (UAS)/Labelled Attachment Score (LAS)) absolute gain. Interestingly, it also supersedes the performance of the state of the art hybrid system by 1.2 points (UAS) absolute gain and shows comparable performance in terms of LAS.¹

CCS Concepts: • Computing methodologies → Natural language processing, neural networks.

Additional Key Words and Phrases: neural dependency parsing, data augmentation, transfer learning, cross-lingual/mono-lingual pretraining, self-training, multi-task learning, low-resourced morphologically rich language: Sanskrit

ACM Reference Format:
Jivnesh Sandhan, Laxmidhar Behera, and Pawan Goyal. 2022. Systematic Investigation of Strategies Tailored for Low-Resource Settings for Sanskrit Dependency Parsing. 1, 1 (January 2022), 13 pages.

1 INTRODUCTION

The Sanskrit Computational Linguistics (SCL) community has garnered growing interest in the dependency parsing task in the last two decades. Earlier attempts for developing dependency parser for the Sanskrit mainly focused on building rule-based systems [Goyal et al. 2007; Kulkarni 2013; Kulkarni et al. 2010, 2019; Kulkarni 2021] due to lack of labelled dataset. With the recent availability of task-specific labelled data, Krishna et al. [2020b,c] propose hybrid systems that integrate a linguistically motivated data-driven system with rules from Pāṇini grammar [Panini BCE] and report the state of the art performance for SDP. These hybrid systems rely on a lexicon driven shallow parser called Sanskrit Heritage Reader (SHR)² for computing linguistically motivated features. SHR does not use a domain-specific lexicon; hence it may fail to recognize some words of an input sentence. In realistic scenarios, failure in generating

¹Code and data will be publicly available at: https://github.com/Jivnesh/SanDP
²https://sanskrit.inria.fr/DICO/reader.fr.html

Authors’ addresses: Jivnesh Sandhan, IIT Kanpur, Kanpur, India, jivnesh@iitk.ac.in; Laxmidhar Behera, IIT Kanpur, Kanpur, India; Pawan Goyal, IIT Kharagpur, Kharagpur, India.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2022 Association for Computing Machinery.

Manuscript submitted to ACM
linguistically motivated features makes these systems handicapped. Further, the morphologically rich nature of the Sanskrit language intensifies the possibility of such failures.

Recently, purely data-driven neural architectures [Dozat and Manning 2017; Fernández-González and Gómez-Rodríguez 2019; Zhou and Zhao 2019] have become de-facto alternative in the Natural Language Processing community due to their ease of applicability and scalability. Thus, Krishna [2019]; Krishna et al. [2020a] investigate the efficacy of these architectures from purely data-driven family to obviate the need of linguistically motivated hand-crafted feature engineering; however, they do not match the performance of hybrid dependency parsers [Krishna et al. 2020b,c] due to data sparsity. Sanskrit being a relatively free word order and a low-resourced morphologically rich language (MRL), demands relatively large labelled data to build robust parsing solution. Human resources for developing additional labelled data are scarce because annotators need to be experts in Sanskrit and native speakers can not do the job [Hellwig et al. 2020].

In this work, we investigate: How far can we push a purely data-driven system using various techniques proposed in low-resource setting [Hedderich et al. 2021]? To answer this question, we systematically explore five pragmatic strategies for low-resource settings, namely, data augmentation [Gulordava et al. 2018; Şahin and Steedman 2018; Vania et al. 2019], cross-lingual/mono-lingual pretraining [Conneau et al. 2020; Kondratyuk and Straka 2019; Peters et al. 2018; Sandhan et al. 2021b], sequential transfer learning [Ruder et al. 2019], multi-task learning [Nguyen and Verspoor 2018] and self-training [Clark et al. 2018; Rotman and Reichart 2019] (§ 3). Our proposed ensembled system outperforms by 2.8/3.9 points (UAS/LAS) absolute gain compared to the state of the art purely data-driven system [Dozat and Manning 2017]. Notably, it also outperforms hybrid system Krishna et al. [2020b] by 1.2 points absolute gain in terms of UAS metric and shows comparable performance in terms of LAS metric (§ 4). Finally, we investigate on the robustness of the proposed system to the relatively free word order nature of Sanskrit. We also perform analysis on what kind of error reduction occurs as a result of effective integration of various strategies (§ 5).

2 BACKGROUND: SANSKRIT DEPENDENCY PARSING

The Pāṇinian framework, the oldest dependency grammar, is generative and deterministically irreversible. That leaves the scope of ambiguity when it is reversed for the Sanskrit analyzer. Goyal et al. [2007] proposed a rule-based system and pointed out the need for a hybrid scheme where the data-driven statistical method is integrated with the rule-based system to resolve ambiguity posed by the irreversibility of grammar rules. Later, Hellwig [2009] took forward this idea and improved performance using a hybrid approach, purely statistical parser augmented with simple syntactic rules from grammar. Earlier attempts for developing dependency parser for the Sanskrit mainly focused on building rule-based systems [Goyal et al. 2007; Kulkarni 2013; Kulkarni et al. 2010, 2019; Kulkarni 2021]. Kulkarni et al. [2010] proposed a graph-based approach where each word is taken as a node and relations between words are identified using grammar rules. Based on heuristics, they rank the exhaustive candidate space. Later, they improved the computational aspect of their method in the following work [Kulkarni 2013]. Earlier attempts for building dependency parser were targeted for simple prose order sentences so Kulkarni et al. [2019] extended their previous work for poetry sentences. However, this system enumerates all possible solutions. Due to the increasing popularity of dependency parser across NLP, Goyal and Kulkarni [2014] introduced a methodology to convert enriched constituency trees into unlabelled dependency tree. Recently, Krishna [2019] proposed structured prediction framework for dependency parsing. This linguistically motivated model uses a graph-based parsing approach and currently reports a state-of-the-art performance for Sanskrit. Although the results of this system are very promising, it takes input from lexicon driven shallow parsers. If parser fails to identify word, this system can not produce a parsed tree. Recently, Sandhan et al. [2021b] proposed
pretraining approach for low-resource dependency parsing for Sanskrit. Due to lack of powerful morphological analyzer, it is challenging to have this information during run time. Therefore, Sandhan et al. [2021b] obviate the need of morphological information during run time with the help of proposed pretraining method. Data-driven approaches have shown tremendous achievement in the field of NLP [Hellwig and Nehrdich 2018; Sandhan et al. 2019]. Specifically, for dependency parsing, neural-based approaches have attracted intense attention of researchers due to state-of-the-art performance without explicit feature engineering [Dozat and Manning 2017; Fernández-González and Gómez-Rodríguez 2019; Zhou and Zhao 2019]. However, Krishna et al. [2020a] reports that these purely data-driven approaches do not match the performance of hybrid counterparts due to data scarcity. Thus, in this work, we investigate the following question: How far can we push a purely data-driven approach using recently proposed strategies for low-resource settings? We experiment with five strategies, namely, data augmentation, sequential transfer learning, pretraining, multi-task learning and self-training. Similar to our work, Vania et al. [2019] also investigate data augmentation, cross-lingual training, and transliteration for low-resource dependency parsing. However, we specifically focus on Sanskrit language with exhaustive experiments on 5 strategies.

3 INVESTIGATION OF STRATEGIES TAILORED FOR LOW-RESOURCE SETTINGS

In this section, we explore five strategies, specially tailored for low-resource settings and ensemble the best performing strategy of each category in our proposed system (§ 3.6). We utilize BiAFFINE parser [Dozat and Manning 2017] as a base system for all the experiments, henceforth referred to as BiAFF.

**Dataset and metric:** We utilize around 4,000 dependency labelled trees from the Sanskrit Treebank Corpus [Kulkarni et al. 2010, STBC] and Śiśupālavadha [Ryali 2016]. We use 1,700, 1,000 and 1,300 sentences as train, dev and test set, respectively. We do these investigations on the dev set to find out the best strategy from each category. Following Krishna et al. [2020c], we use sentence level macro averaged Unlabelled and Labelled Attachment Scores (UAS, LAS).

### 3.1 Data Augmentation

| System               | UAS  | LAS  |
|----------------------|------|------|
| BiAFF                | 84.07| 76.87|
| Mixed [Krishna et al. 2020c] | 64.71| 53.99|
| Cropping [Şahin and Steedman 2018] | 71.48| 66.43|
| Rotation [Şahin and Steedman 2018] | 83.76| 76.54|
| Nonce [Gulordava et al. 2018] | 84.74| 77.83|
| Nonce++ [Gulordava et al. 2018] | 84.67| 77.47|

Table 1. Results on dev set when different data augmentation strategies augmented with BiAFF.

**Systems:** Table 1 reports data augmentation strategies experimented for SDP. Mixed: Krishna et al. [2020c] use mixture of existing multiple augmentation techniques like synonym replacement [Zhang et al. 2015], sentence simplification [Vickrey and Koller 2008], sentence cropping [Şahin and Steedman 2018] and linguistically motivated constraints to generate augmented data. Şahin and Steedman [2018] introduce Cropping: deletes some parts of a sentence to create multiple short meaningful sentences and Rotation: permutes the siblings of headword restricted to a set of relations. Both operations modify a set of words or configurational information; however, they do not change the dependencies. Nonce: Gulordava et al. [2018] propose to create nonce sentences by substituting a few words which share the same
syntactic labels. For creating each training instance, stochastically, they replace content words with the words having
the same part of speech tag, morphological tag and dependency label as syntactic constraints.\textsuperscript{3} However, they pick
replacement pairs from the same training data, which may not be better choice to reduce OOV issues. \textbf{Nonce++}: Thus,
on top of the \textbf{Nonce} setting, we experiment with replacement pairs other than training data. To reduce OOV, we use
the rule-based parser [Kulkarni 2013] on unlabelled data for obtaining potential candidates for replacement which also
satisfy the syntactic constraints.

\textbf{Observations}: Except \textbf{Nonce}/\textbf{Nonce++} from Table 1, all systems show low performance compared to BiAFF trained
with gold data. \textbf{Rotation} seems useful for modelling data in a poetry domain; since our data is in a prose domain,
permutations in word order resulted in results in loss of potentially useful configurational information for prose domain.
\textbf{Cropping} possibly shows performance drop of 12.5/10.4 points (UAS/LAS) due to the generation of shorter training
samples relative to test data. Similarly, the low performance of \textbf{Mixed} might be due to presence of the cropping
technique and lack of syntactic constraints while using synonym replacement. On the other hand, \textbf{Nonce} setting
shows significant improvement over BiAFF, possibly due to the hard syntactic constraints it follows. Without altering
configuration information and hampering sentence length, \textbf{Nonce} shows significant improvement over BiAFF. \textbf{Nonce++}
does not improve further.

\subsection{3.2 Cross/mono-lingual Pretraining}

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|}
\hline
System & UAS & LAS \\
\hline
BiAFF [Dozat et al. 2017] & 84.07 & 76.87 \\
BiAFF+mBERT [Kondratyuk and Straka 2019] & 80.29 & 69.65 \\
BiAFF+ELMo [Peters et al. 2018] & 84.00 & 76.78 \\
BiAFF+XLM-R [Nguyen et al. 2021] & 86.02 & 78.84 \\
BiAFF+LCM [Sandhan et al. 2021a] & 86.28 & 80.17 \\
\hline
\end{tabular}
\caption{Results on dev set when BiAFF is integrated with cross-lingual/mono-lingual pretraining strategies.}
\end{table}

\textbf{Systems}: We briefly elaborate on the pretraining approaches reported in Table 2. Due to bottleneck of unlabelled
data for Sanskrit, training monstrous contextual pretraining approaches from scratch may not be helpful; hence, the
natural choice is multilingual pretraining approach. Thus, we experiment with two multilingual pretraining approaches,
namely, multilingual BERT [Devlin et al. 2019, \textbf{mBERT}] based system [Kondratyuk and Straka 2019] and XLM-Roberta
[Conneau et al. 2020, \textbf{XLM-R}] based system [Nguyen et al. 2021]. Also, we investigate on \textbf{ELMo} [Peters et al. 2018]
trained from scratch for Sanskrit [Sandhan et al. 2021a]. Also, we evaluate pretraining, which use morphological tagging
related auxiliary tasks [Sandhan et al. 2021b, \textbf{LCM}].

\textbf{Observations}: \textbf{mBERT} shows 3.8/7.2 points (UAS/LAS) drop over BiAFF (Table 2) due to absence of Sanskrit language
during pretraining [Sandhan et al. 2021b]. On the other hand, \textbf{XLM-R} (with Sanskrit during pretraining) demonstrates
1.9/2.0 points absolute gain (UAS/LAS) over BiAFF. On par performance of \textbf{ELMo} trained from scratch might be
attributed to insufficient unlabelled data. Finally, [Sandhan et al. 2021b, \textbf{LCM}], with the help of morphological enriched

\footnote{Following Vania et al. [2019], we additionally consider dependency label as a constraint.}

Manuscript submitted to ACM
encoders shows 2.2/3.3 points (UAS/LAS) absolute gain.\textsuperscript{4} Notably, gains obtained for LAS metric is higher by 1.1 points which helps to bridge the gap between UAS-LAS metrics.

### 3.3 Self-training

| System             | UAS  | LAS  |
|--------------------|------|------|
| BiAFF [Dozat et al. 2017] | 84.07 | 76.87 |
| Self-Train         | 84.15 | 77.03 |
| CVT [Clark et al. 2018] | 82.40 | 73.62 |
| DCST [Rotman and Reichart 2019] | \textbf{85.61} | \textbf{78.85} |

Table 3. Results on dev set for self-training based systems.

**Systems:** Another line of modelling focuses on self-training [Clark et al. 2018; Goldwasser et al. 2011; Rybak and Wróblewska 2018] to overcome the bottleneck of task-specific labelled data. Earlier attempts failed to prove effectiveness of self-training for dependency parsing [Rush et al. 2012]. However, [Clark et al. 2018, CVT] and Rotman and Reichart [2019][DCST], show successful application, thus, we consider these two systems. Also, we generate dependency data by applying a pretrained BiAFF system on unlabelled data. We augment this predicted data with gold data and retrain BiAFF in Self-Train setting.

**Observation:** As shown in Table 3, DCST outperforms due to its syntactically enriched contextual representations. The default setting of CVT does not consider the potentially useful morphological information for the SDP task. Thus, it falls short in terms of performance. Self-Train does not show significant improvement over BiAFF.

### 3.4 Sequential Transfer Learning

| System         | UAS  | LAS  |
|----------------|------|------|
| BiAFF          | 84.07 | 76.87 |
| SeqTraL-FE     | 80.03 | 73.60 |
| SeqTraL-FEA    | 82.92 | 76.51 |
| SeqTraL-UF     | 84.00 | 77.72 |
| SeqTraL-DL     | 84.78 | 78.53 |
| SeqTraL-FT     | \textbf{84.84} | \textbf{78.53} |

Table 4. Results on dev set for Transfer Learning using various optimization schemes.

**Systems:** Transfer Learning has captivated the attention of many NLP researchers due to it’s wide emergence and ease of integration. It contains two stages, namely, pretraining and adaptation. In pretraining, general representation is learned from related source tasks, and in the latter stage, learned knowledge is used for target tasks. However, the selection of pretraining and target task is interrelated. Sequential Transfer Learning (SeqTraL) consists of two components, namely, pretraining for learning general representations and adaptation to facilitate sample efficiency [Ruder et al. 2019]. We investigate on adapting the pretrained morphological tagger trained on the same morphologically tagged dependency data.\textsuperscript{5} Here, we use the first three Bi-LSTM layers from this morphological tagger and integrate

\textsuperscript{4}In contrast to Sandhan et al. [2021b], we also use oracle morphological information as an input.

\textsuperscript{5}Sequence tagger consists of similar LSTM-based encoder as [Dozat and Manning 2017] and decoder with fully connected layer followed by softmax layer.
them in the BiAFF and investigate with various optimization schemes, proposed for reducing a catastrophic forgetting [French 1999; McCloskey and Cohen 1989]. As we move down, in Table 4, the freedom for adaptation increases. Table 4 reports baselines used to investigate the best optimization scheme to adopt pretrained encoder. **SeqTraL-FE:** We treat newly integrated layers as Feature Extractors (FE) by freezing them. **SeqTraL-FEA:** In the SeqTraL-FE, we augment adaptor modules [Houlsby et al. 2019; Stickland and Murray 2019] in between newly added consecutive layers. **SeqTraL-UF:** Gradually Unfreeze (UF) these new layers in the top to down order [Felbo et al. 2017; Howard and Ruder 2018]. **SeqTraL-DL:** The discriminative learning rate (DL) is used for newly added layers [Howard and Ruder 2018], the learning rate is decreased from top-to-bottom layers. **SeqTraL-FT:** We fine tune (FT) all the newly added layers with the default learning rate.

**Observations:** Table 4 shows that performance improves as the freedom for adaption increases. The SeqTraL-FT (fine-tuned) reports the best result with 1-2 points absolute gain over BiAFF.

### 3.5 Multi-task Learning

| Tasks      | UAS  | LAS  |
|------------|------|------|
| BiAFF      | 84.07| 76.87|
| MTL-Case   | 84.81| 77.47|
| MTL-Label  | 84.87| 77.88|
| MTL-Morph  | **84.84**| **78.00**|

Table 5. Results on dev set for different auxiliary tasks in multi-task learning setting.

**Systems:** Multi-task learning helps to exploit the complementary signal from related auxiliary tasks which enables better generalization. We simultaneously train BiAFF and a sequence labelling based auxiliary task in a multi-task setting (MTL). We experiment with the following auxiliary tasks: prediction of the morphological label (MTL-Morph), dependency relation between a word and its head (MTL-Label) and the case label (MTL-Case).

**Observations:** Table 5 illustrates that all auxiliary tasks show significant gains over BiAFF and MTL-Morph marks highest gains. We also experiment with a setting where all three tasks are considered simultaneously; however, this shows subpar performance compared to MTL-Morph.

### 3.6 The Proposed Ensembled System

| System        | UAS  | LAS  |
|---------------|------|------|
| BiAFF         | 84.07| 76.87|
| + SeqTraL-FT  | 84.84| 78.53|
| + LCM         | 86.22| 79.93|
| + MTL-Morph   | **86.55**| **80.30**|
| + Nonce       | 85.91| 79.68|
| + DCST        | 85.86| 79.46|

Table 6. Ablation analysis on dev set for different components of the proposed system.

---

6 We use POS label in the absence of case information.

Manuscript submitted to ACM
Ablation: Table 6 reports the contribution of different components for the proposed system. We observe consistent improvements with SeqTraL-FT, LCM and MTL-Morph integrated on the top of BiAFF, due to their complementary signals. The most prominent contribution comes from the morphologically tailored pretraining method [Sandhan et al. 2021b, LCM] which confirms the well-established findings of the effectiveness of morphological information for dependency parsing. However, DCST and Nonce augmentation strategy do not add any complementary signal; hence, we do not consider them in the proposed system.

The proposed system: Figure 1 shows the ensembled proposed system using a toy example from Sanskrit. It consists of two steps, namely, pretraining (LCM) and integration. As shown in Figure 1a, LCM pretrains three encoders $E^{(1)}$–$E^{(3)}$ using three independent auxiliary tasks, namely, morphological label prediction, case label prediction and relation label prediction. Thereafter, as shown in Figure 1b, these pretrained encoders are integrated with the BiAFF encoder $E^{(P)}$ using a gating mechanism as employed in Sato et al. [2017]. We use SeqTraL-FT optimization scheme to update the weights of these four encoders. Next, MTL-Morph component adds morphological tagging as an auxiliary task to inject complementary signal in the model. Finally, the combined representation of a pair of words in passed to BiAFF to calculate probability of arc score (S) and label (L).

4 EXPERIMENTS

Data and Metric: We utilize around 4,000 dependency labelled trees from the Sanskrit Treebank Corpus [Kulkarni et al. 2010, STBC] and Śīśupālavadha [Ryali 2016]. We use 1,700, 1,000 and 1,300 sentences as train, dev and test set, respectively. However, the final results on the test set are reported using systems trained with combined gold train and
dev set. Additionally, we also evaluate on the recently proposed Vedic Sanskrit Treebank [Hellwig et al. 2020, VST] consisting of 2,524 and 1,473 sentences as training and test data, respectively. The STBC contains sentences from prose domain alone; however, VST contains sentences from poetry and prose domain. The STBC follows annotations based on Kāraka [Kulkarni et al. 2010; Kulkarni and Sharma 2019], the grammatical tradition of Sanskrit, while the VST uses Universal Dependency (UD). Following Krishna et al. [2020c], we use sentence level macro averaged Unlabelled and Labelled Attachment Scores (UAS, LAS) and t-test for statistical significance [Dror et al. 2018].

**Baselines:** There are two broad approaches proposed for the dependency parsing task, namely, transition-based and graph-based parsing. We use More et al. [2019][YAP] and Chang et al. [2016][L2S] from transition-based dependency parsing family. Dozat and Manning [2017][BiAFF] is a graph-based approach with BiAFFINE attention mechanism. Krishna et al. [2020b][MG-EBM] extends Krishna et al. [2020c][Tree-EBM-F] using multi-graph formulation. We report their standalone numbers for fair comparison. Systems marked with (*) are hybrid systems which leverage linguistic rules from Pāṇinian grammar.

**Hyper-parameters:** We use default hyper-parameters for all the competing baseline systems compared in this work. For SeqTraL variants, we use the exact same encoder as Ma et al. [2018] with 2 Bi-LSTM layers and decoder with fully connected layer followed by softmax layer. For the proposed system, we adopt BiAFF’s codebase by Ma et al. [2018] with the hyper-parameters setting as follows: the batch size as 16, training iterations as 100, a dropout rate as 0.33, the number of stacked Bi-LSTM layers as 2, learning rate as 0.002 and the remaining parameters as the same as Ma et al. [2018].

| System      | STBC UAS | STBC LAS | VST UAS | VST LAS |
|-------------|----------|----------|---------|---------|
| YAP         | 75.31    | 66.02    | 70.37   | 39.20   |
| L2S         | 81.97    | 74.14    | 72.44   | 62.76   |
| Tree-EBM-F  | 82.65    | 79.28    | -       | -       |
| BiAFF       | 85.88    | 79.55    | 77.23   | 67.68   |
| Tree-EBM-F* | 85.32    | 83.93    | -       | -       |
| MG-EBM*     | 87.46    | 84.70    | -       | -       |
| Ours        | 88.67    | 83.47    | 79.71   | 69.89   |

Table 7. Main results on test set for SDP. Hybrid systems, marked with (*) are not directly comparable with our system. Our results are statistically significant compared to the strong baseline BiAFF as per t-test (p < 0.01). Results are averaged over 3 runs.

**Results:** Clearly, the proposed ensembled system outperforms the state of the art purely data-driven system (BiAFF) by 2.8/3.9 points (UAS/LAS) absolute gain.7 Interestingly, it also supersedes the performance of the hybrid state of the art system [Krishna et al. 2020b, MG-EBM] by 1.2 points (UAS) absolute gain and shows comparable performance for LAS metric. We observe that performance of transition-based systems (YAP/L2S) is significantly low compared to graph-based counterparts (BiAFF/Ours). We also obtain a similar performance trend for VST data. The VST data is a mixture of dependency labelled trees from both poetry and prose domain. As a result, the overall performance for VST

---

7The hybrid systems, EBM* and MG-EBM* use extra-linguistic knowledge; hence, they are not directly comparable.

Manuscript submitted to ACM
is low compared to STBC due to loss of configurational information.  

5 ERROR ANALYSIS

![Graphs showing performance analysis](image)

Fig. 2. (a) Performance on prose and poetry domain as sentence length increases. The dependency label-wise performance on (b) in vocabulary (c) out of vocabulary words. From left to right the label frequency increases. Performance in terms of (d) dependency length (e) distance to root and (f) degree of non-projectivity.

| Domain | BiAFF UAS | BiAFF LAS | Ours UAS | Ours LAS |
|--------|-----------|-----------|----------|----------|
| Prose  | 85.88     | 79.55     | 88.67    | 83.47    |
| Poetry | 47.75     | 44.32     | 52.86    | 50.45    |
| In Vocab | 86.25 | 81.76     | 89.16    | 85.64    |
| OOV    | 79.01     | 69.97     | 80.37    | 72.26    |

Table 8. Fine-grained analysis in between the strong baseline: BiAFF and the proposed system on test set.

We compare the proposed system with the strong baseline BiAFF.  
We investigate the robustness of systems based on the following language specific phenomena such as (1) relatively free word order nature: Here, we verify the robustness to the configurational information by evaluating on poetry and prose domains. Table 8 illustrates that the proposed system shows consistently superior performance over BiAFF; however, both systems are brittle to poetry domain; on an average the performance degrades by 36.8/34.0 points (UAS/LAS) compared to prose counterpart. The

---

8We do not evaluate Tree-EBM-F* and MG-EBM* on VST data due to the unavailability of the codebase.
9We could not compare with Tree-EBM-F* and MG-EBM* due to unavailability of codebase and system outputs.

Manuscript submitted to ACM
prose/poetry data has 44.0/0.6% word pairs with dependency length as 1, 13.5/30.7% word pairs with dependency length more than 5 and 7.1/19.2% non-projective arc. Clearly, these structural divergences explain the low performance on poetry domain due to systems trained on prose domain. (2) morphologically rich nature: is investigated by focusing on out of vocabulary phenomenon. Being a morphologically rich language, the OOV phenomenon is crucial for modelling robust systems. The test set of STBC consists of 34% words that are OOV. The prominent contribution to average drop (reported in Table 8) of 7.9/12.6 points (UAS/LAS) comes from failure to predict rarely occurring dependency relation, corresponding to OOV words correctly. In order to verify the performance in OOV vocabulary scenario, we evaluate the dependency label-wise performance for out of vocabulary (OOV) words (Figure 2c) and in vocabulary words (Figure 2b). The contrast between Figure 2b and 2c demonstrates that both systems perform poorly when a word is OOV and dependency label is rarely occurring.10

Figure 2a illustrates the empirical performance in terms of sentence length to validate the robustness of the proposed system against varied sentence lengths. We notice downwards trend in the performance as the sentence length increases. Particularly, both systems drastically go down in poetry domain. Following Kulmizev et al. [2019]; McDonald and Nivre [2011], we analyze the performance in terms of dependency length11 (Figure 2d), distance to root12 (Figure 2e) and degree of non-projectivity13 (Figure 2f). If we contrast between the performance of both the systems in poetry and prose domain then they show the declined performance in terms of sentence length, distance to root and degree of non-projectivity. In Figure 2d, we observe slightly improved performance for both the systems in prose domain as the dependency length increases. This can be attributed to ability of graph-based parsers to capture long range dependencies well. Figure 2 only reports the performance on labelled score; however, we observe a similar trend for unlabelled score. Summarily, we conclude that both systems perform similarly in the all aspects of error analysis where the proposed system shows consistent improvements.

6 CONCLUSION AND DISCUSSION
We focused on dependency parsing for low-resourced morphologically rich Sanskrit language. To tackle limitations of the existing engineering-based approaches, we investigate the efficacy of recently proposed strategies specially tailored for low-resource settings. Our results showed that the proposed ensembled system significantly improves with 2.8/3.9 points (UAS/LAS) compared to BiAFF. Interestingly, it also superseded the performance of state of the art hybrid system MG-EBM∗ by 1.2 points (UAS) absolute gain and showed comparable performance in terms of LAS metric. We plan to extend this work by investigating on ways to make current system robust for poetry domain.

ACKNOWLEDGMENTS
We thank Amha Kulkarni for providing Sanskrit dependency treebank data, Anupama Ryali for Śīsūpālavadha dataset and the anonymous reviewers for their constructive feedback towards improving this work. The work of the first author is supported by the TCS Fellowship under the Project TCS/EE/2011191P.

10Opposite observation for the frequency marked with 1 is due to the fact that the total number of candidates belonging to this label class is very less so systems are not able to generalize well for this class.
11The dependency length is defined as the distance between head-dependent pair when arranged linearly in a sentence.
12The distance of a node to its root in a dependency tree.
13The degree of non-projectivity for a head(x)/dependent(y) pair is defined as the number of words occurrences in between x and y which are not part of decedents of x and modify a word outside x and y window.
REFERENCES

Kai-Wei Chang, He He, Hal Daumé, John Langford, and Stefanie Ross. 2016. A Credit Assignment Compiler for Joint Prediction. In Proceedings of the 30th International Conference on Neural Information Processing Systems (Barcelona, Spain) (NIPS’16). Curran Associates Inc., Red Hook, NY, USA, 1713–1721.

Kevin Clark, Minh-Thang Luong, Christopher D. Manning, and Quoc Le. 2018. Semi-Supervised Sequence Modeling with Cross-View Training. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Brussels, Belgium, 1914–1925. https://doi.org/10.18653/v1/D18-1217

Alexis Conneau, Kartikay Khadwal, 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. Association for Computational Linguistics, Online, 8440–8451. https://doi.org/10.18653/v1/2020.acl-main.747

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). Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

Timothy Dozat and Christopher D. Manning. 2017. Deep Biaffine Attention for Neural Dependency Parsing. In 5th International Conference on Learning Representations. ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net. https://openreview.net/forum?id=Hk95PK9le

Timothy Dozat, Peng Qi, and Christopher D Manning. 2017. Stanford’s graph-based neural dependency parser at the conll 2017 shared task. In Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. 20–30.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The Hitchhiker’s Guide to Testing Statistical Significance in Natural Language Processing. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Melbourne, Australia, 1383–1392. https://doi.org/10.18653/v1/P18-1128

Bjarke Felbo, Alan Mislove, Anders Sagaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Copenhagen, Denmark, 1615–1625. https://doi.org/10.18653/v1/D17-1169

Daniel Fernández-Gonzalez and Carlos Gómez-Rodriguez. 2019. Left-to-Right Dependency Parsing with Pointer Networks. 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). Association for Computational Linguistics, Minneapolis, Minnesota, 710–716. https://doi.org/10.18653/v1/N19-1076

Robert M. French. 1999. Catastrophic forgetting in connectionist networks. Trends in cognitive sciences 3, 4 (1999), 128–135.

Dan Goldwasser, Roi Reichart, James Clarke, and Dan Roth. 2011. Confidence Driven Unsupervised Semantic Parsing. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, Portland, Oregon, USA, 1486–1495. https://www.aclweb.org/anthology/P11-1149

Pawan Goyal, Vipul Arora, and Lakshmiha Behera. 2007. Analysis of Sanskrit text: Parsing and semantic relations. In Sanskrit Computational Linguistics. Springer, 200–218.

Pawan Goyal and Amba Kulkarni. 2014. Converting phrase structures to dependency structures in Sanskrit. In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers. 1834–1843.

Kristina Golordava, Piotr Bojanowski, Edouard Grave, Täl 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). Association for Computational Linguistics, New Orleans, Louisiana, 1195–1205. https://doi.org/10.18653/v1/N18-1108

Michael A. Hedderich, Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. 2021. A Survey on Recent Approaches for Natural Language Processing in Low-Resource Scenarios. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, Online, 2545–2568. https://doi.org/10.18653/v1/2021.naacl-main.287

Oliver Hellwig. 2009. Extracting Dependency Trees from Sanskrit Texts. In Sanskrit Computational Linguistics, Third International Symposium, Hyderabad, India, January 15-17, 2009. Proceedings (Lecture Notes in Computer Science, Vol. 5406), Amba P. Kulkarni and Gérard P. Huët (Eds.). Springer, 106–115. https://doi.org/10.1007/978-3-540-93885-9_9

Oliver Hellwig and Sebastian Neheduc. 2018. Sanskrit Word Segmentation Using Character-level Recurrent and Convolutional Neural Networks. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Brussels, Belgium, 2754–2763. https://doi.org/10.18653/v1/D18-1295

Oliver Hellwig, Salvatore Scarlata, Elia Ackermann, and Paul Widmer. 2020. The Treebank of Vedic Sanskrit. In Proceedings of the 12th Language Resources and Evaluation Conference. European Language Resources Association, Marseille, France, 5137–5146. https://aclanthology.org/2020.lrec-1.632

Neil Houlsby, Andrei Giurgu, Stanisław Jastrzębski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-Efficient Transfer Learning for NLP. In Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97). Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, Long Beach, California, USA, 2790–2799.
Panini. 500 BCE. Ashtadhyayi. http://panini.phil.hhu.de/panini/panini/

Jeremy Howard and Sebastian Ruder. 2018. Universal Language Model Fine-tuning for Text Classification. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Melbourne, Australia, 328–339. https://doi.org/10.18653/v1/P18-1031

Dan Kondratyuk and Milan Straka. 2019. 75 Languages, 1 Model: Parsing Universal Dependencies Universally. 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). Association for Computational Linguistics, Hong Kong, China, 2779–2795. https://doi.org/10.18653/v1/D19-1279

Amrith Krishna. 2019. Addressing Language Specific Characteristics for Data-Driven Modelling of Lexical, Syntactic and Prosodic Tasks in Sanskrit. In Ph.D. thesis, IIT Kharagpur, Kharagpur, West Bengal, India.

Amrith Krishna, Ashim Gupta, Deepak Garasangi, Jwesh Sandhan, Pavankumar Satuluri, and Pawan Goyal. 2020a. Neural Approaches for Data Driven Dependency Parsing in Sanskrit. arXiv:2004.08076 [cs.CL]

Amrith Krishna, Ashim Gupta, Deepak Garasangi, Pavankumar Satuluri, and Pawan Goyal. 2020b. Keep it Surprisingly Simple: A Simple First Order Graph Based Parsing Model for Joint Morphosyntactic Parsing in Sanskrit. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, Online, 4791–4797. https://doi.org/10.18653/v1/2020.emnlp-main.388

Amba Kulkarni. 2013. A deterministic dependency parser with dynamic programming for Sanskrit. In Proceedings of the Second International Conference on Computational Linguistics (Depling 2013). 157–166.

Amba Kulkarni, Sheetal Polkar, and Devanand Shulki. 2010. Designing a constraint based parser for Sanskrit. In International Sanskrit Computational Linguistics Symposium. Springer, 70–90.

Amba Kulkarni and Dipthi Sharma. 2019. Paninian Syntactico-Semantic Relation Labels. In Proceedings of the Fifth International Conference on Dependency Linguistics (Depling, SyntaxFest 2019). Association for Computational Linguistics, Paris, France, 198–208. https://doi.org/10.18653/v1/W19-3724

Amba Kulkarni, Sanal Vikram, and Sriram K. 2019. Dependency Parser for Sanskrit Verses. In Proceedings of the 6th International Sanskrit Computational Linguistics Symposium. Association for Computational Linguistics, IIT Kharagpur, India, 14–27. https://www.aclweb.org/anthology/W19-7582

Amba P. Kulkarni. 2021. Sanskrit Parsing Following Indian Theories of Verbal Cognition. Transactions on Asian and Low-Resource Language Information Processing 20 (2021), 1 – 38.

Artur Kulmizev, Miryam de Lhoneux, Johannes Gontrum, Elena Fano, and Joakim Nivre. 2019. Deep Contextualized Word Embeddings in Transition-Based and Graph-Based Dependency Parsing - A Tale of Two Parsers Revisited. 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). Association for Computational Linguistics, Hong Kong, China, 2755–2768. https://doi.org/10.18653/v1/D19-1277

Xuezhe Ma, Zecong Hu, Jingzhou Liu, Nanyun Peng, Graham Neubig, and Eduard Hovy. 2018. Stack-Pointer Networks for Dependency Parsing. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Melbourne, Australia, 1403–1414. https://doi.org/10.18653/v1/P18-1130

Michael McCloskey and Neal J. Cohen. 1989. Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. In Psychology of Learning and Motivation, Gordon H. Bower (Ed.). Vol. 24. Academic Press, 109 – 165. https://doi.org/10.1016/S0079-7421(08)60536-8

Ryan McDonald and Joakim Nivre. 2011. Analyzing and Integrating Dependency Parsers. Computational Linguistics 37, 1 (2011), 197–230. https://doi.org/10.1162/coli_a_00839

Amir More, Amit Seker, Victoria Basmova, and Reut Tsarfaty. 2019. Joint Transition-Based Models for Morpho-Syntactic Parsing: Parsing Strategies for MRLs and a Case Study from Modern Hebrew. Transactions of the Association for Computational Linguistics 7 (March 2019), 33–48. https://doi.org/10.1162/tacl_a_00253

Dat Quoc Nguyen and Karin Verspoor. 2018. An Improved Neural Network Model for Joint POS Tagging and Dependency Parsing. In Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. Association for Computational Linguistics, Brussels, Belgium, 81–91. https://doi.org/10.18653/v1/K18-2008

Minh Van Nguyen, Viet Duc Lai, Amir Pouran Ben Veyseh, and Thien Huu Nguyen. 2021. Trankit: A Light-Weight Transformer-based Toolkit for Multilingual Natural Language Processing. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations.

Panini. 500 BCE. Ashtadhyayi. http://panini.phil.hhu.de/panini/panini/
Sebastian Ruder, Matthew E. Peters, Swabha Swayamdipta, and Tomas Wolf. 2019. Transfer Learning in Natural Language Processing. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials. Association for Computational Linguistics, Minneapolis, Minnesota, 15–18. https://doi.org/10.18653/v1/N19-5004

Alexander Rush, Roi Reichart, Michael Collins, and Amir Globerson. 2012. Improved Parsing and POS Tagging Using Inter-Sentence Consistency Constraints. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Association for Computational Linguistics, Jeju Island, Korea, 1344–1444. https://www.aclweb.org/anthology/D12-1131

Anupama Ryali. 2016. CHALLENGES IN DEVELOPING SANSKRIT E-READERS: SEMI-AUTOMATICALLY USING ONLINE ANALYSER SAMSADHANI WITH SPECIAL REFERENCE TO SISUPALAVADHA OF MAGHA. In Workshop on Bridging the Gap Between Sanskrit CL Tools Management of Sanskrit DL, Article 5 (July 2016). http://cse.iitkgp.ac.in/resgrp/cnerg/sclws/papers/ryali.pdf

Alexander Rush, Roi Reichart, Michael Collins, and Amir Globerson. 2012. Improved Parsing and POS Tagging Using Inter-Sentence Consistency Constraints. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Association for Computational Linguistics, Jeju Island, Korea, 1344–1444. https://www.aclweb.org/anthology/D12-1131

Anupama Ryali. 2016. CHALLENGES IN DEVELOPING SANSKRIT E-READERS: SEMI-AUTOMATICALLY USING ONLINE ANALYSER SAMSADHANI WITH SPECIAL REFERENCE TO SISUPALAVADHA OF MAGHA. In Workshop on Bridging the Gap Between Sanskrit CL Tools Management of Sanskrit DL, Article 5 (July 2016). http://cse.iitkgp.ac.in/resgrp/cnerg/sclws/papers/ryali.pdf

Alexander Rush, Roi Reichart, Michael Collins, and Amir Globerson. 2012. Improved Parsing and POS Tagging Using Inter-Sentence Consistency Constraints. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Association for Computational Linguistics, Jeju Island, Korea, 1344–1444. https://www.aclweb.org/anthology/D12-1131

Anupama Ryali. 2016. CHALLENGES IN DEVELOPING SANSKRIT E-READERS: SEMI-AUTOMATICALLY USING ONLINE ANALYSER SAMSADHANI WITH SPECIAL REFERENCE TO SISUPALAVADHA OF MAGHA. In Workshop on Bridging the Gap Between Sanskrit CL Tools Management of Sanskrit DL, Article 5 (July 2016). http://cse.iitkgp.ac.in/resgrp/cnerg/sclws/papers/ryali.pdf

Piotr Rybak and Alina Wróblewska. 2018. Semi-supervised neural system for tagging, parsing and lematization. In Proceedings of the CoNLL 2018 Shared Task: Multilingual parsing from raw text to universal dependencies. 45–54.

Gözde Gül Şahin and Mark Steedman. 2018. Data Augmentation via Dependency Tree Morphing for Low-Resource Languages. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Brussels, Belgium, 5004–5009. https://doi.org/10.18653/v1/D18-1545

Jivnesh Sandhan, Om Adideva, Digumarthi Komal, Laxmidhar Behera, and Pawan Goyal. 2021a. Evaluating Neural Word Embeddings for Sanskrit. arXiv:2104.00270 [cs.CL]

Jivnesh Sandhan, Om Adideva, Digumarthi Komal, Laxmidhar Behera, and Pawan Goyal. 2021a. Evaluating Neural Word Embeddings for Sanskrit. arXiv:2104.00270 [cs.CL]

Jivnesh Sandhan, Om Adideva, Digumarthi Komal, Laxmidhar Behera, and Pawan Goyal. 2021a. Evaluating Neural Word Embeddings for Sanskrit. arXiv:2104.00270 [cs.CL]

Jivnesh Sandhan, Amrith Krishna, Pawan Goyal, and Laxmidhar Behera. 2019. Revisiting the Role of Feature Engineering for Compound Type Identification in Sanskrit. In Proceedings of the 6th International Sanskrit Computational Linguistics Symposium. Association for Computational Linguistics, IIT Kharagpur, India, 28–44. https://www.aclweb.org/anthology/W19-7503

Jivnesh Sandhan, Amrith Krishna, Ashish Gupta, Laxmidhar Behera, and Pawan Goyal. 2021b. A Little Pretraining Goes a Long Way: A Case Study on Dependency Parsing Task for Low-resource Morphologically Rich Languages. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop. Association for Computational Linguistics, Online, 111–120. https://aclanthology.org/2021.eacl-srw.16

Motoki Sato, Hitoshi Manabe, Hiroshi Noji, and Yuji Matsumoto. 2017. Adversarial Training for Cross-Domain Universal Dependency Parsing. In Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. Association for Computational Linguistics, Vancouver, Canada, 71–79. https://doi.org/10.18653/v1/K17-3007

Asa Cooper Stickland and Iain Murray. 2019. BERT and PALs: Projected Attention Layers for Efficient Adaptation in Multi-Task Learning. In Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97). Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, Long Beach, California, USA, 5986–5995. http://proceedings.mlr.press/v97/stickland19a.html

Clara Vania, Yova Kementchedjhieva, Anders Søgaard, and Adam Lopez. 2019. A systematic comparison of methods for low-resource dependency parsing on genuinely low-resource languages. 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). Association for Computational Linguistics, Hong Kong, China, 1105–1116. https://doi.org/10.18653/v1/D19-1102

David Vickrey and Daphne Koller. 2008. Sentence Simplification for Semantic Role Labeling. In Proceedings of ACL-08: HLT. Association for Computational Linguistics, Columbus, Ohio, 344–352. https://aclanthology.org/P08-1040

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level Convolutional Networks for Text Classification. In Advances in Neural Information Processing Systems, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (Eds.), Vol. 28. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2015/file/258cf8b51c773f3f8dcd84be867a9a02-Paper.pdf