Supplementary Material for Keep It Surprisingly Simple: A Simple First Order Graph Based Parsing Model for Joint Morphosyntactic Parsing in Sanskrit

Amrith Krishna[1], **Ashim Gupta**[2], **Deepak Garasangi**[3]*,
Pavankumar Satuluri[4] **and Pawan Goyal**[5]
[1]Department of Computer Science and Technology, University of Cambridge
[2]School of Computing, University of Utah [3]Independent Researcher
[4]School of Linguistics & Literary Studies, Chinmaya Vishwavidyapeeth
[5]Department of Computer Science and Engineering, IIT Kharagpur
ak2329@cam.ac.uk, ashim@cs.utah.edu, pawang@cse.iitkgp.ac.in

1 Morphologically Informed Pruning

Cconsider a 14 word verse, "*Śriyaḥ patiḥ śrīmati śāsituṁ jagat jagannivāsaḥ vasudevasadmani vasan dadarśa avatarantam ambarāt hiraṇyagarbhāṅgabhuvam munim hariḥ*"[1], from the literary work '*Śiśupālavadha*'. Here, the sequence is in its segmented form, and yet it would result in 23,040 different possible sentences owing to syncretism and homonymy.[2] The analysis for the given sequence is shown in Figure 1. Further, each of the sentence would result in n^{n-2} unlabelled spanning trees if we consider an unpruned complete graph as the input. Here n is the number of tokens in the sentence, i.e. 14 for the given sentence. This will result in a prohibitively large space of possible spanning trees, if we consider the possible spanning trees for all the 23,040 possible sentences. However, with our linguistic pruning we could restrict the number of total possible spanning trees to be just about 102,360 trees (combined count for all the 23,040 sentences).

As discussed in the main paper, an input multigraph is formed from the morphological analysis. Spanning trees are enumerated from the input multigraph and for each spanning tree, we evaluate its validity as a candidate dependency tree. Based on the edge selected from the input multigraph, every node in the tree will have a specific morphological tag. Further, we will assume the edges to be labelled with dependency relations, but only with those which are applicable as per the morphological tags of the nodes in the edge. If the tree evaluates to be a valid candidate, then its edges are retained in the multigraph. However, the label information from the tree is not added to the multigraph. The unlabelled pruned multigraph forms the input to MG-EBM.

Figure 2 shows the dependency tree for the aforementioned sequence. Here, "hariḥ" is assigned as the subject (kartā) and "munim" as the object (karma), with the main verb "dadarśa" as the common head for both. The morphological analysis for "hariḥ" produces two possible analyses, both of them in nominative case[3]. Similarly, morphological analysis for "munim" produces two analyses, both in accusative case. Let us now illustrate some cases of using linguistic information to validate the candidacy of the generated spanning tree as a candidate dependency tree.

1. Consider the edges from "dadarśa" to "hariḥ" and "dadarśa" to "munim" in the generated tree. Here, since "hariḥ" is in nominative case, it can either be assigned the subject (kartā) relation or the object relation (karma). Being in the accusative case, "munim" on the other hand, can only take the object relation (karma). If we assume "hariḥ" becomes the karma, instead of kartā, then the tree will eventually evaluate to false, as the edge between "munim" and "dadarśa" cannot be assigned any label. In such a case the tree is not a valid candidate tree. Though the previous state of the art model, T-EBM*, also perform linguistically informed pruning, it makes the decisions greedily, by considering only 3 nodes at a time. Here, we cannot check if all the edges will eventually have a label.

2. The linguistically informed pruning can at best be seen as a rule-based deterministic delexicalised dependency parsing approach. For instance, if two nodes, connected by an

*Work done while at IIT Kharagpur

[1] The sentence translates to, Lakṣmi's consort,Viṣṇu, who is the source of the world, who was born in the house of Vasudeva to control the world, saw Brahma's son Nārada, descending from the sky.

[2] https://bit.ly/3eAmr9G

[3] Both tags differ by gender which is inconsequential here

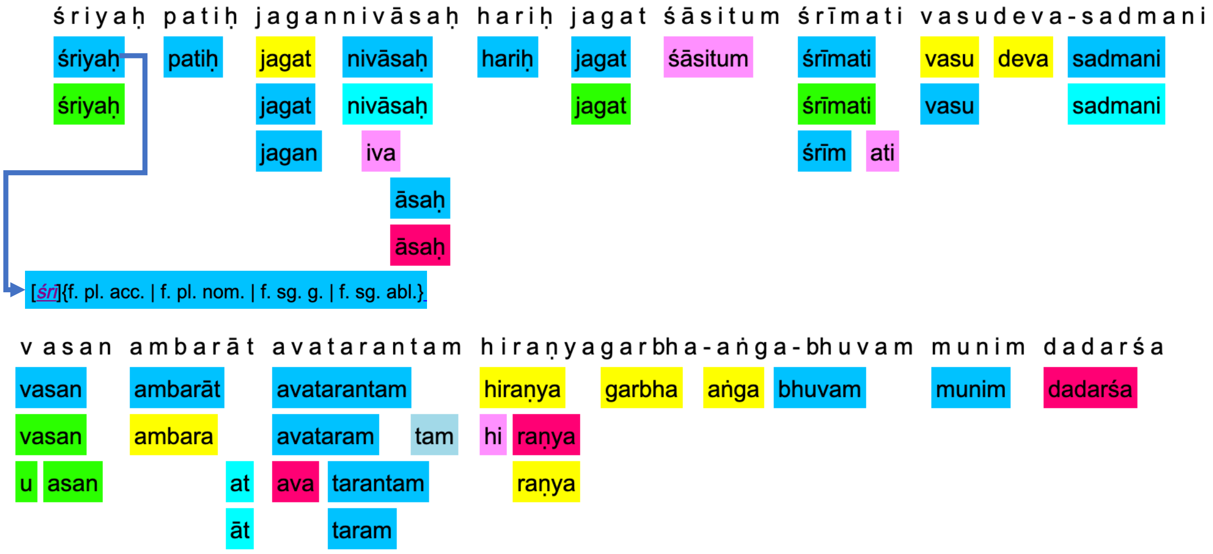

Figure 1: The morphological analysis from the lexicon driven shallow parser for the given input sequence. The cases of syncretism for the word *Śriyaḥ* is also shown. In our analysis we assume that the sequence is segmented and each token is a word. Hence, we do not consider the cases where the tokens are split by the analyser, such as “u” and “asan” instead of “vasan”

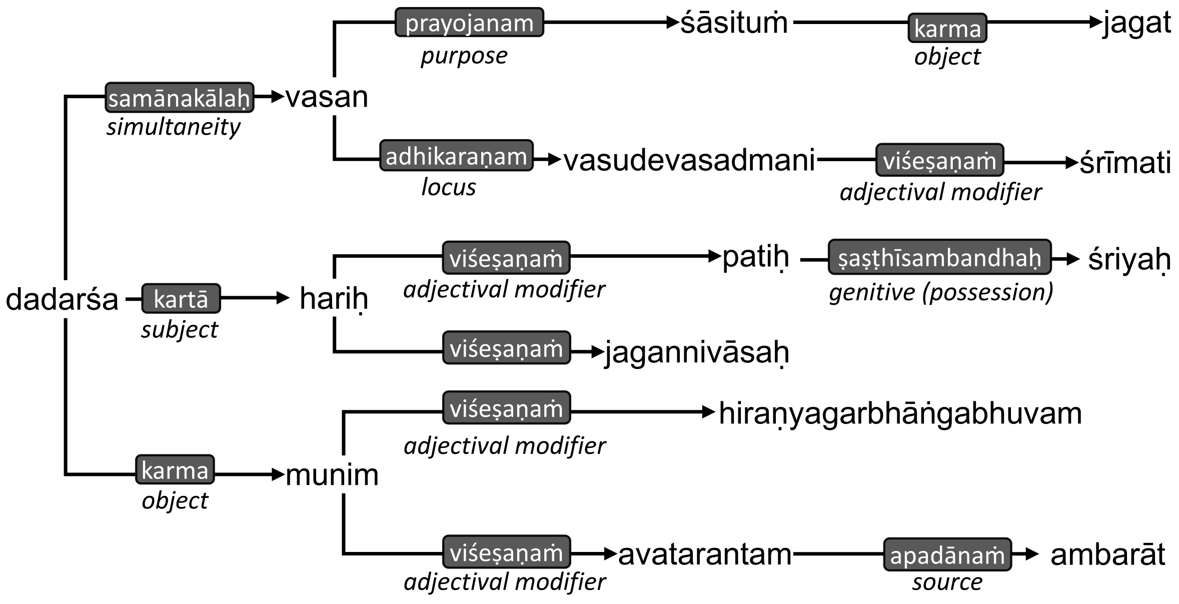

Figure 2: Dependency analysis for the sequence in ‘Śiśupālavadha’. The corresponding English translation for the dependency translation are given beneath each of these edge labels.

edge, have to be assigned a viśeṣaṇa (adjectival modifier) relation, then both the words must agree on all the three grammatical categories of a noun, i.e. case, number and gender. Now, the word “patiḥ” is an adjective to “hariḥ”. Since both the words are in nominative case, our pruning approach will not be able to decide on which of the two to be assigned the head.[4] If we assume “patiḥ” as the head, it will be eligible to be connected to the verb with kartā relation, instead of “hariḥ”. Summarily, our pruning approach would validate both the candidate trees, i.e. one where hariḥ is assigned as the child of the kartā relation, as well as where patiḥ is assigned as the child of the kartā relation. To disambiguate between these, we require a data-driven approach, like MG-EBM, where the feature set makes use of distributional information of lemma and surface forms as well.

3. The word “ambarāt” has only one analysis and it is in ablative case. It is is connected to “avatarantam” with the “apadānam” relation in its gold dependency tree (Figure 2). Here, ablative case words can be attached to a verb to form either a “hetu” or an “apadānam” relation. It needs to be noted that the word “avatarantam” is a nominal, though it is assigned as the head for a relation which is intended for a verb as the head. However, in Sanskrit, derivational nouns derived from a verb, are often considered as a verb when assigning the relations. We consider such cases as well in our pruning.

2 DCST++: Neural Morphosyntactic Parser

Neural Multi-task Morphological Parser (MTL): We first give an overview of the multi-task morphological parser, that forms backbone for DCST++. Gupta et al. (2020) propose a multi-task neural tagger, MTL, for morphological tagging in Sanskrit. Here, the composite morphological tag of a word is broken down into multiple grammatical categories, and a hierarchy is established between the categories. Each category is considered a separate task and then a model is trained in a multi-task setting. They provide some evidence that there might be an implicit hierarchy among grammatical categories in Sanskrit. For example, they found that the grammatical category *number* benefits from supervision at the shallowest layer, whereas *tense* benefits from supervision at deeper layers. The parser when evaluated on our test set reports a sentence level macro averaged F1-score of 63.55 %.

DCST++ We now briefly describe DCST++, which integrates MTL with the DCST model (Rotman and Reichart, 2019). DCST or the Deep Contextualised Selft Training Parser essentially extends the biaffine parser by Dozat and Manning (2017) with self training. Here, first a base parser is trained with limited data, which is the biaffine parser from Dozat and Manning (2017). Then, dependency parses for unlabelled sentences are obtained using the base parser. The predicted trees are then used as input for training various sequence level auxiliary tasks. The auxiliary tasks performed in the paper are, predicting the number of children for each node in the tree, the distance of each node from the root and the relative POS encoding of each word in the tree. For these sequence level tasks, the input tree is adapted to a suitable sequence tagging scheme. Then the main parser is trained, where the biaffine parser is retrained with a fresh initialisation. However this time, the encoders from the auxiliary tasks are used to generate representations for the input and all of these representations are combined using a gating mechanism as proposed by Sato et al. (2017). DCST++ predicts only the dependency parsing results, and does not predict the morphological parsing results.

Table 1 shows the dependency parsing results, where none of the three DCST configurations use gold morphological tags as input. The neural DCST parser results reported in the main paper used surface form and gold morphological tags as input. Here, we report the results of the DCST configuration where the surface form and a coarse level POS tag from a rule based POS tagger is used as input. The tagset contains 4 POS tags, namely, nominal, finite verb, infinite verb and indeclinable. The model reports a UAS and LAS of 80.8 and 70.99, respectively. DCST + MTL is a pipeline model, where the standard DCST model uses predicted morphological tags from MTL as input. Here, we can observe that DCST+MTL and DCST++ report a UAS of 81.62 and 81.73, respectively.

[4] In a Sanskrit sentence, written in prose, the adjective of a nominal would always precede it. However, this need not be true for sentences in verse form. We do not make any assumptions on the writing style of the input.

Model	**UAS**	**LAS**
DCST	80.80	70.99
DCST + MTL	81.62	71.64
DCST++	**81.73**	**72.28**

Table 1: **Sentence-level macro UAS and LAS**, Comparison of Neural Dependency Parsers

Dataset: We use a test set of 1,300 sentences, where 1,000 come from Sanskrit Tree Bank Corpus (STBC) (Kulkarni, 2013) and 300 from the Sisupāla-vadha (Ryali, 2016). 1,500 and 1,000 sentences from STBC, other than the ones in the test data, were used as the training and validation data, respectively for DCST, DCST++, and BiAFF. However all the EBM models and YAP were trained on 12,320 sentences obtained by augmenting the training data in STBC (Krishna et al., 2020, §4.1). Here, the authors applied synonym replacement (Zhang et al., 2015), sentence simplifications (Vickrey and Koller, 2008) and sentence cropping approaches for the augmentation (Sahin and Steedman, 2018). BiAFF, DCST and DCST++ performed worse, when used with the sentences from the augmented training data. The dataset and the evaluation code can be downloaded at `http://bit.ly/KISSData`

References

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.

Ashim Gupta, Amrith Krishna, Pawan Goyal, and Oliver Hellwig. 2020. Evaluating neural morphological taggers for sanskrit.

Amrith Krishna, Bishal Santra, Ashim Gupta, Pavankumar Satuluri, and Pawan Goyal. 2020. A graph based framework for structured prediction tasks in sanskrit. *Computational Linguistics*, (accepted).

Amba Kulkarni. 2013. A deterministic dependency parser with dynamic programming for Sanskrit. In *Proceedings of the Second International Conference on Dependency Linguistics (DepLing 2013)*, pages 157–166, Prague, Czech Republic. Charles University in Prague, Matfyzpress, Prague, Czech Republic.

Guy Rotman and Roi Reichart. 2019. Deep contextualized self-training for low resource dependency parsing. *Transactions of the Association for Computational Linguistics*, 7:695–713.

Anupama Ryali. 2016. Challenges in developing sanskrit e-readers semi-automatically using online analyzer sasādhanī with special reference to śiśupālavadha of māgha. In *Workshop on Bridging the Gap Between Sanskrit CL Tools Management of Sanskrit DL, ICON2016*.

Gozde Gul Sahin 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*, pages 5004–5009, Brussels, Belgium. Association for Computational Linguistics.

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*, pages 71–79, Vancouver, Canada. Association for Computational Linguistics.

David Vickrey and Daphne Koller. 2008. Sentence simplification for semantic role labeling. In *Proceedings of ACL-08: HLT*, pages 344–352, Columbus, Ohio. Association for Computational Linguistics.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 649–657. Curran Associates, Inc., Montreal, Canada.