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

Abstract

Morphologically rich languages seem to benefit from joint processing of morphology and syntax, as compared to pipeline architectures. We propose a graph-based model for joint morphological parsing and dependency parsing in Sanskrit. Here, we extend the Energy based model framework (Krishna et al., 2020), proposed for several structured prediction tasks in Sanskrit, in 2 simple yet significant ways. First, the framework's default input graph generation method is modified to generate a multigraph, which enables the use of an exact search inference. Second, we prune the input search space using a linguistically motivated approach, rooted in the traditional grammatical analysis of Sanskrit. Our experiments show that the morphological parsing from our joint model outperforms standalone morphological parsers. We report state of the art results in morphological parsing, and in dependency parsing, both in standalone (with gold morphological tags) and joint morphosyntactic parsing setting.

1 Introduction

Morphology and syntax are often inextricably intertwined for morphologically rich languages (MRLs). For such languages, it might be unrealistic to design dependency parsers that expect correct morphological tags to be provided as input (More et al., 2019; Bohnet et al., 2013). Jointly modelling *morphological parsing* (MP) with *dependency parsing* (DP) has shown to be effective for several MRLs (More et al., 2019). In this work, we present *multigraph-EBM* (MG-EBM), a joint model for morphosyntactic parsing, i.e. joint MP and DP, in Sanskrit.

Morphosyntactic parsing has been successfully applied to several MRLs. Bohnet et al. (2013) proposed a transition based joint parser, extending the joint POS tagger and dependency parser of Bohnet and Nivre (2012). Similarly Seeker and Çetinoğlu (2015) proposed a joint graph based parser for Turkish. Here, two different models, one predicting a morphological path and the other a dependency tree, are made to reach an agreement using dual decomposition. More et al. (2019) proposed a transition based joint parser for Hebrew, where it aims to maximise a global score over both morphological and dependency transitions.

Sanskrit is an MRL which shows high degree of syncretism and homonymy in its morphological paradigm. About 90.96 % of the tokens in a dataset of 115,000 Sanskrit sentences (Krishna et al., 2017) show syncretism with an average of 3.62 morphological tags per token. Morphological features, especially *case*, are indicative of the syntactic roles that a word (nominal) can assume in a sentence. The interplay of morphological markers and syntactic roles has been formalised in the traditional grammatical analysis of Sanskrit, *Ashtādhyāyī* (Pāṇini, 500 BCE; Ramkrishnamacharyulu, 2009). Here, the joint modelling of syntactic and morphological information can help disambiguate each other (Tsarfaty, 2006). In MG-EBM, we use this information to prune the input search space.

Krishna et al. (2020) proposed an energy based-model (EBM) framework for multiple structured prediction tasks in Sanskrit. For all the models under EBM, the input search space is a graph which considers every unique morphological analysis of the input words to be a separate node (Figure 1a). Modeling morphosyntactic parsing over this input graph requires an approximation algorithm for inference as it needs to predict a structure containing only a subset of the nodes. We propose to modify the input space to be a multigraph where the number of nodes correspond to the number of tokens in a sentence. This enables us to use an exact search inference (Edmonds, 1967) akin to the first

*Work done while at IIT Kharagpur

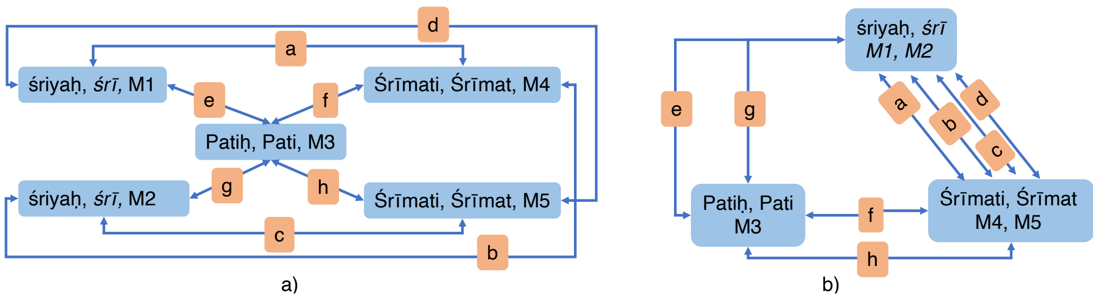

Figure 1: Input graph for the Sanskrit sequence, "śriyaḥ patiḥ śrīmati" in a) Original EBM configuration: every unique triple, *(surface-form, stem, morphological-tag)* forms a separate node. b) MG-EBM configuration: All the nodes with the same surface form are merged, while retaining all the edges in their original configuration.The edge labels in both the figures show that all the edges are retained. The labels should not be confused as dependency relations. The graphs we use as input to the EBM are unlabelled graphs.

order dependency parsing model of McDonald et al. (2005).

MG-EBM achieves state of the art results (SOTA), improving the previous best results by 3 F-score points for MP and 2 UAS points for standalone DP (expects gold morph tags as input). We set new SOTA results for joint MP and DP. Further, we demonstrate that MP results obtained from our joint model outperforms standalone MP models. Our proposed pruning approach in itself report an improvement of 6 F-score (2 UAS) points with the original EBM configuration for MP (DP), and a further 2 F-Score (2 UAS) points improvement with the multigraph formulation for MP (DP).

2 Energy Based Model

Krishna et al. (2020) proposed an Energy based model (EBM) framework (LeCun et al., 2006) for multiple structured prediction tasks in Sanskrit. The framework is a generalisation of the joint word-segmentation (WS) and morphological parsing (MP) model by Krishna et al. (2018). The models under this framework are trained using multilayer perceptrons and are essentially first-order arc-factored graph-based parsing models. The dependency parsing (DP) model, similar to McDonald et al. (2005), makes use of a sequence level max-margin loss (Taskar et al., 2003) and Chu-Liu-Edmonds algorithm for the inference. However, the feature function for the task is learnt automatically, and differs from that of McDonald et al. (2005). A lexicon-driven shallow parser (Goyal and Huet, 2016; Huet, 2005) is used to enumerate all the morphological analyses, including cases of syncretism and homonymy, for the tokens in the input sequence. An input graph is constructed from this analysis, as shown in Figure 1a for the sequence "*śriyaḥ patiḥ śrīmati*".[1] Here every node is a unique combination of three entities, namely, surface-form, stem and morphological tag. All the node pairs, which are not suggested as alternative solutions and hence can co-occur in a predicted solution, form an edge.

The framework uses an automated feature learning approach (Lao and Cohen, 2010; Meng et al., 2015) to generate a feature function consisting of 850 features. Using this feature set the framework achieves state of the art (SOTA) results in several tasks. This is significant, given several morphologically rich languages still rely on models that use hand-crafted features for SOTA results (More et al., 2019; Seeker and Çetinoğlu, 2015). Given the models are arc-factored, the edges are featurised. A feature would consider only one entity each from either of the nodes in the edge. The feature then calculates the distributional information between these entities conditioned on some specific morphological constraint. The type of the entities and the constraints, which constitute the features, are automatically learned as typed paths over a large morphologically tagged corpus. While the training and the feature function remain the same for all the tasks under the framework, the inference is task specific. It searches for a spanning tree with minimum energy (Edmonds, 1967) for DP. For tasks that require prediction of a subset of nodes from the input graph, such as WS and MP, and standalone MP, approximation algorithms were used for inference. Here the inference procedure searches for only a small percentage of all the possible candidates ($< 1\%$) (Krishna et al., 2018).

[1]gloss: *śriyaḥ* - Of godess Lakshmi, *patiḥ* - husband, *śrīmati* - prosperous

3 MG-EBM: The Proposed Model

Multigraph-EBM (MG-EBM) extends the EBM framework in two simple yet significant ways.

Multigraph formulation: Instead of the 'one node per unique morphological analysis', as shown in Figure 1a, we propose to use a 'one node per inflected surface-form' multigraph representation, as shown in Figure 1b. For instance, the surface form śriyaḥ, due to syncretism, has 2 possible morphological analyses *M1* and *M2*. In Figure 1a, these analyses are represented as separate nodes and are connected to the only analysis of *patiḥ* via the edges *e* and *g*. In Figure 1b, the cases of syncretism for *śriyaḥ* are merged as a single node, though the edges e and g to *patiḥ* are retained. This leads to a multigraph formulation.[2] The new representation retains all the edges, and their feature vectors, present in the original representation. The design of our feature function guarantees that every edge will have a unique feature vector. With this formulation, we simplify the search problem for the joint MP and DP task to that of searching for the spanning tree with minimum energy. This enables the use of the exact search Edmonds-Chu-Liu MST algorithm (Edmonds, 1967), rather than an approximation algorithm, for inference. It is straightforward to extend the algorithm to multigraph, as we just need to retain only the minimum energy edge and prune out all the other edges between a pair of nodes in the input graph (McDonald and Satta, 2007).

Linguistically Motivated Pruning: Linguistic constraints based on the traditional grammatical analysis and verbal cognition in Sanskrit (Kulkarni and Ramakrishnamacharyulu, 2013; Ramkrishnamacharyulu, 2009) have been previously employed in various deterministic dependency parsers for Sanskrit (Kulkarni et al., 2019; Kulkarni, 2013; Kulkarni et al., 2010). We use these constraints to prune the edges in our input graph. During pruning, we first exhaustively enumerate all the unlabelled directed spanning-trees in the input graph using Mayeda and Seshu (1965).[3] For each such tree, if every directed-edge in the spanning-tree can be assigned at least one label as per the rules of the grammar, it will be considered a valid candidate. Finally, all the edges which are not part of even one valid candidate-tree will be pruned from the input graph. This pruned unlabelled directed graph serves as the input for the inference procedure. This linguistically informed pruning can at best be seen as a rule-based deterministic delexicalised dependency parsing, which considers information only from the morphological tags.

Morphology can signal dependency in various ways. The morphological marker of a word may not only index the properties of the word itself, but it may also index the agreement between its head or dependants (Nichols, 1986). The agreement between the subject and verb in terms of the number and person respectively is one such case. Similarly, the case, number, and gender agreement between the words in an adjectival modifier (*viśeṣaṇa*) relation is another example of this in Sanskrit. Further, morphological markers are indicative not only of the presence of syntactic dependency between the words in a sentence, but also of the type of the syntactic dependency shared between them (Nichols, 1986; Seeker and Kuhn, 2013). In Sanskrit, the *case* information of a nominal narrows down the possible relations it can have with a verb as the head. This is shown in Figure 2. We form constraints based on these morphosyntactic information and use it for pruning the edges.[4] The dependency relations shown in Figure 2 are rooted in the *Pāṇinian* grammar, i.e. traditional grammatical sys-

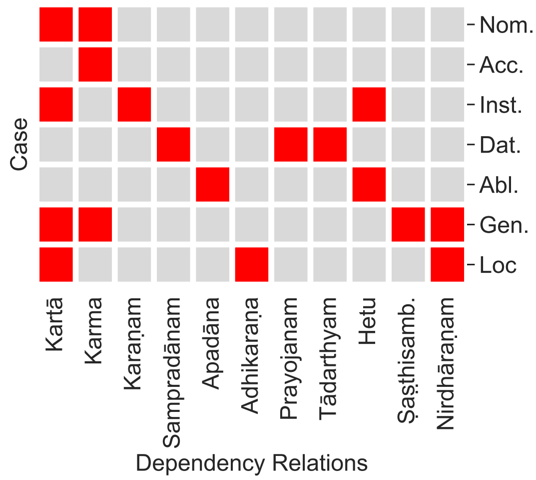

Figure 2: Case information and possible dependency relations that a case is indicative of.

[2] It needs to be noted that the edge labels in Figure 1(a and b) are used for illustrative purposes. The graphs we use are unlabelled and do not contain any information related to dependency labels.

[3] The algorithm has an amortised runtime of $O(nm)$, where n and m are the number of nodes and edges, respectively (Smith, 1997).

[4] Refer to supplementary material §1 for example cases

System	UAS	LAS
YAP	76.99	73.02
BiAff	82.35	75.65
DCST	84.36	76.8
T-EBM	82.65	79.28
T-EBM*	85.32	83.93
MG-EBM	**87.46**	**84.70**

(a) Standalone dependency parsing results

	System	Morph F1	UAS/F1	LAS/F1
Standalone MP	P-EBM	84.36	-	-
	C-EBM	93.89	-	-
Pipeline	C-EBM+T-EBM*	93.89	- / 77.22	- / 72.87
Neural	DCST++	-	81.73	71.90
Joint MP and DP	JP-EBM*	89.59	- / 79.2	- / 74.03
	JP-EBM-Prune	95.21	- /81.28	- / 75.84
	MG-EBM	**97.14**	**83.16**	**79.33**

(b) Results for joint morphosyntactic parsing and morphological parsing

Table 1: Results for: a) DP when the gold morphological tags are provided as input. b) MP and DP when no gold morphological tags are provided as input. JP-EBM*, JP-EBM-Prune and pipeline models are reported in F-Score.

tem followed for Sanskrit (Pāṇini, 500 BCE). The use of the relations from the *Pāṇinian* grammar, instead of other dependency tagsets such as Universal Dependencies (Nivre et al., 2016), enables us to incorporate the linguistic constraints for the pruning.

4 Experimental Framework

Systems: For DP, with gold morphology tags as input, we compare the performance of MG-EBM with 5 other parsers. The models are YAP (More et al., 2019): a transition based parser for MRLs, BiAff (Dozat and Manning, 2017): a neural biaffine classifier, DCST (Rotman and Reichart, 2019): a self-training based neural classifier, and two variants of EBM: T-EBM and T-EBM*. For MP, we use the current SOTA model, C-EBM, an EBM variant as the baseline. The EBM variants and the impact of these variations would be elaborated in Section 5. For the joint morphosyntactic setting, we propose DCST++ as a neural baseline. DCST++ is our augmentation over DCST which integrates encoder outputs from a neural morphological tagger (Gupta et al., 2020) by a gating mechanism (Sato et al., 2017).[5]

Metric: All the results we report are macro averaged at a sentence level. For DP, we use UAS and LAS and for MP, we use F-Score. For joint MP and DP, all the EBM models other than MG-EBM may predict a tree that has a different vertex set than that of the ground truth. Since UAS cannot be used here, we use (unlabelled and labelled) F-Score for those systems. For MG-EBM UAS/LAS and Unlabelled/Labelled F-Score would be the same.

Dataset[6]: We use a test set of 1,300 sentences, where 1,000 come from the Sanskrit Tree Bank Corpus (Kulkarni, 2013, STBC) and 300 from *Sisupāla-vadha*, a work from classical Sanskrit poetry (Ryali, 2016). 1,500 and 1,000 sentences from STBC, other than the ones in 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).[7]

5 Results

MG-EBM achieves the state of the art (SOTA) results in MP and in DP, both in standalone (with gold morphological tags as input) and joint morphosyntactic parsing setting. Table 1a shows that MG-EBM reports a 2 point improvement in UAS as compared to T-EBM*, the previous SOTA model for DP in Sanskrit. Both MG-EBM and T-EBM* differ only in terms of how the pruning of the input graph is performed. The pruning decisions in T-EBM* are made by considering a maximum of 3 nodes at a time, instead of the context from the entire tree. However, MG-EBM considers a tree in its entirety for applying the constraints. This leads to more than 300 fold reduction in the number of possible candidates for MG-EBM as compared to T-EBM*, with just about 40 % increase in wall time (on test data).[8] Since, both the models use the same label predictor (Krishna et al., 2020), they perform similar, with a small improvement of 0.77 points for MG-EBM .

Table 1b shows that MG-EBM outperforms C-EBM, the previous SOTA model for morphologi-

[5]Refer to the supplementary material §2 for more experiments with this model

[6]The dataset can be downloaded from `http://bit.ly/KISSData`

[7]BiAFF, DCST and DCST++ performed worse, when used with the sentences from the augmented training data.

[8]The increase is due to the use of the spanning tree enumeration algorithm by Mayeda and Seshu (1965).

cal parsing. Similarly MG-EBM achieves SOTA results for DP in the joint setting, followed by DCST++. In the joint setting, gold morphological tags are not provided as input. All the EBM models, other than MG-EBM, use the one node per analysis (Figure 1a) input formulation and approximation algorithms for inference. For morphological parsing, the inference in C-EBM searches for a maximal clique, considering pairwise interaction between all the nodes in the clique, while P-EBM searches for a Steiner Tree. Both JP-EBM* and JP-EBM-Prune extend P-EBM for joint morphosyntactic parsing, by introducing linguistically informed pruning. JP-EBM* uses the same pruning approach as T-EBM*, while JP-EBM-Prune uses our proposed pruning approach. The models report a 5 point and 11 point F-Score increase respectively for morphological parsing as compared to P-EBM. In fact, JP-EBM-Prune outperforms C-EBM. MG-EBM and JP-EBM-Prune use the same pruning approach proposed in this work. They differ in terms of the input space formulation and as a consequence, MG-EBM uses an exact search inference. This difference has led to nearly 2 point increase in both UAS and F-Score, and a 3 Point increase in LAS between both. YAP (More et al., 2019), the SOTA joint morphosyntactic parser proposed originally for Hebrew can perform joint prediction. However it is observed that YAP's performance would typically degrade in the joint setting as compared to its performance in the standalone setting (with gold-morphological tag; Table 1a). All the joint models for morphological parsing and DP outperform YAP even when YAP uses gold morphological tags. Finally, all the joint models for morphological parsing and DP outperform the pipeline EBM model C-EBM + T-EBM*, which validates that joint morphosyntactic parsing benefits an MRL like Sanskrit than a pipeline model.

6 Conclusion

In this work, we proposed MG-EBM, a model for joint morphological parsing and DP in Sanskrit. It extends the EBM framework from Krishna et al. (2020) by 1) incorporating a linguistically motivated pruning approach resulting in a substantial reduction in the input search space, and 2) modifying the input graph formation to a multigraph resulting in the use of Edmonds-Chu-Liu algorithm (Edmonds, 1967), an exact search algorithm, as inference. While the multigraph formulation is language agnostic the linguistically motivated pruning is rooted on the grammatical tradition of Sanskrit. Experiments validate that the joint morphosyntactic-parsing hypothesis, i.e., morphological information can benefit syntactic disambiguation and vice versa (Tsarfaty, 2006), holds true for Sanskrit. We find that the MG-EBM reports state of the art results (SOTA) for morphological parsing, outperforming standalone morphological parsing models, similar to what is observed for Hebrew (More et al., 2019). Further, all the joint morphological parsing and DP variants of EBM, we experimented here, result in a superior performance than the pipeline morphological parsing and DP EBM model. We also establish SOTA results in Sanskrit for DP, both in standalone and joint setting.

Acknowledgements

We thank Dr Amba Kulkarni for helpful comments and discussions regarding the work. Further, we thank Dr Kulkarni for sharing the STBC data with us. We thank Dr Anupama Ryali for sharing Sisupāla-vadha dataset with us.

References

Bernd Bohnet and Joakim Nivre. 2012. A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1455–1465, Jeju Island, Korea. Association for Computational Linguistics.

Bernd Bohnet, Joakim Nivre, Igor Boguslavsky, Richárd Farkas, Filip Ginter, and Jan Hajič. 2013. Joint morphological and syntactic analysis for richly inflected languages. *Transactions of the Association for Computational Linguistics*, 1:415–428.

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.

Jack Edmonds. 1967. Optimum branchings. *Journal of Research of the national Bureau of Standards B*, 71(4):233–240.

Pawan Goyal and Gerard Huet. 2016. Design and analysis of a lean interface for sanskrit corpus annotation. *Journal of Language Modelling*, 4(2):145–182.

Ashim Gupta, Amrith Krishna, Pawan Goyal, and Oliver Hellwig. 2020. Evaluating neural morphological taggers for sanskrit. In *Proceedings of the 17th*

Workshop on Computational Research in Phonetics, Phonology, and Morphology. Association for Computational Linguistics.

Gérard Huet. 2005. A functional toolkit for morphological and phonological processing, application to a Sanskrit tagger. *Journal of Functional Programming*, 15(4):573–614.

Amrith Krishna, Bishal Santra, Sasi Prasanth Bandaru, Gaurav Sahu, Vishnu Dutt Sharma, Pavankumar Satuluri, and Pawan Goyal. 2018. Free as in free word order: An energy based model for word segmentation and morphological tagging in Sanskrit. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2550–2561, Brussels, Belgium. Association for Computational Linguistics.

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).

Amrith Krishna, Pavan Kumar Satuluri, and Pawan Goyal. 2017. A dataset for Sanskrit word segmentation. In *Proceedings of the Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 105–114, Vancouver, Canada. Association for Computational Linguistics.

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.

Amba Kulkarni, Sheetal Pokar, and Devanand Shukl. 2010. Designing a constraint based parser for sanskrit. In *Sanskrit Computational Linguistics*, pages 70–90, Berlin, Heidelberg. Springer Berlin Heidelberg.

Amba Kulkarni and KV Ramakrishnamacharyulu. 2013. Parsing sanskrit texts: Some relation specific issues. In *Proceedings of the 5th International Sanskrit Computational Linguistics Symposium. DK Printworld (P) Ltd.*

Amba Kulkarni, Sanal Vikram, and Sriram K. 2019. Dependency parser for Sanskrit verses. In *Proceedings of the 6th International Sanskrit Computational Linguistics Symposium*, pages 14–27, IIT Kharagpur, India. Association for Computational Linguistics.

Ni Lao and William W. Cohen. 2010. Relational retrieval using a combination of path-constrained random walks. *Mach. Learn.*, 81(1):53–67.

Yann LeCun, Sumit Chopra, Raia Hadsell, Marc'Aurelio Ranzato, and Fu-Jie Huang. 2006. A tutorial on energy-based learning. In *Predicting Structured Data*. MIT Press.

W. Mayeda and S. Seshu. 1965. Generation of trees without duplications. *IEEE Transactions on Circuit Theory*, 12(2):181–185.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 523–530, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Ryan McDonald and Giorgio Satta. 2007. On the complexity of non-projective data-driven dependency parsing. In *Proceedings of the Tenth International Conference on Parsing Technologies*, pages 121–132, Prague, Czech Republic. Association for Computational Linguistics.

Changping Meng, Reynold Cheng, Silviu Maniu, Pierre Senellart, and Wangda Zhang. 2015. Discovering meta-paths in large heterogeneous information networks. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, pages 754–764, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

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*, 7:33–48.

Johanna Nichols. 1986. Head-marking and dependent-marking grammar. *Language*, 62(1):56–119.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association (ELRA).

Pāṇini. 500 BCE. *Ashtādhyāyī*.

K. V. Ramkrishnamacharyulu. 2009. Annotating sanskrit texts based on śābdabodha systems. In *Sanskrit Computational Linguistics*, pages 26–39, Berlin, Heidelberg. Springer Berlin Heidelberg.

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*.

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.

Wolfgang Seeker and Özlem Çetinoğlu. 2015. A graph-based lattice dependency parser for joint morphological segmentation and syntactic analysis. *Transactions of the Association for Computational Linguistics*, 3:359–373.

Wolfgang Seeker and Jonas Kuhn. 2013. Morphological and syntactic case in statistical dependency parsing. *Computational Linguistics*, 39(1):23–55.

Malcolm James Smith. 1997. *Generating spanning trees*. MS Thesis, University of Victoria.

Ben Taskar, Carlos Guestrin, and Daphne Koller. 2003. Max-margin markov networks. In *Proceedings of the 16th International Conference on Neural Information Processing Systems*, NIPS'03, pages 25–32, Cambridge, MA, USA. MIT Press.

Reut Tsarfaty. 2006. Integrated morphological and syntactic disambiguation for modern Hebrew. In *Proceedings of the COLING/ACL 2006 Student Research Workshop*, pages 49–54, Sydney, Australia. Association for Computational Linguistics.