A Falta de Pan, Buenas Son Tortas:[*]
The Efficacy of Predicted UPOS Tags for Low Resource UD Parsing

Mark Anderson
Universidade da Coruña, CITIC
Department of CS & IT
m.anderson@udc

Mathieu Dehouck
LaTTiCe, CNRS, ENS
Université Sorbonne Nouvelle
mathieu.dehouck@udc.es

Carlos Gómez-Rodríguez
Universidade da Coruña, CITIC
Department of CS & IT
carlos.gomez@udc.es

Abstract

We evaluate the efficacy of predicted UPOS tags as input features for dependency parsers in lower resource settings to evaluate how treebank size affects the impact tagging accuracy has on parsing performance. We do this for real low resource universal dependency treebanks, artificially low resource data with varying treebank sizes, and for very small treebanks with varying amounts of augmented data. We find that predicted UPOS tags are somewhat helpful for low resource treebanks, especially when fewer fully-annotated trees are available. We also find that this positive impact diminishes as the amount of data increases.

1 Introduction

Low resource parsing is a long-standing problem in NLP and many techniques have been introduced to tackle it (Hwa et al., 2005; Zeman and Resnik, 2008; Ganchev et al., 2009; McDonald et al., 2011; Agić et al., 2016). For an extensive review and comparison of techniques see Vania et al. (2019). Here we focus on the utility of part-of-speech (POS) tags as features for low resource dependency parsers.

POS tags are a common feature for dependency parsers. Tiedemann (2015) highlighted the unrealistic performance of low resource parsers when using gold POS tags in a simulated low resource setting. The performance difference was stark despite using fairly accurate taggers, which is not a reasonable assumption for low resource languages. Tagging performance in low resource settings is still very weak even when utilising cross-lingual techniques and other forms of weak supervision (Kann et al., 2020). Even when more annotated data is available, it isn't clear how useful POS tags

are for neural dependency parsers, especially when utilising character embeddings (Ballesteros et al., 2015; de Lhoneux et al., 2017). Work investigating the utility of POS tags typically observe a small increase in performance or no impact when used as features for neural dependency parsers. Smith et al. (2018) found that universal POS (UPOS) tags offer a marginal improvement for their transition based parser for multi-lingual universal dependency (UD) parsing. Dozat et al. (2017) also observed an improvement in parsing performance for graph-based parsers when the predicted UPOS tags came from sufficiently accurate taggers.

Zhang et al. (2020) only found POS tags to be useful for English and Chinese when utilising them as an auxiliary task in a multi-task system. Anderson and Gómez-Rodríguez (2020) found that a prohibitively high accuracy was needed to utilise predicted UPOS tags for both graph- and transition-based parsers for UD parsing. They also obtained results that suggested smaller treebanks might be able to directly utilise less accurate UPOS tags. We evaluate this further by analysing the impact of tagging accuracy on UD parsing in low resource contexts, with regards to the amount of data available to train taggers and parsers.

2 Methodology

We performed three experiments. The first is an evaluation of predicted tags as features for biaffine parsers for real low resource treebanks. It also includes parsers trained with UPOS tagging as an auxiliary task similar to the experiments in Zhang et al. (2020). The second experiment evaluates the impact of different tagging accuracies on different dataset sizes using artificial low resource treebanks by sampling from high resource treebanks. The last experiment utilises a data augmentation technique to investigate the efficacy of predicted UPOS tags for very small treebanks (∼20 sentences) when

---

[*] *Lacking yeast-proven bread, a flatbread alternative will suffice*, i.e. if you can't get more fully-annotated dependency trees, annotating UPOS tags can still be helpful.

augmented with varying amounts of data.

Low resource data   We take all UD v2.6 tree-banks (Zeman et al., 2020) with less than 750 sentences in both its training dataset and development dataset. We cluster these treebanks into two groups, very low with less than 50 sentences and low with less than 750. The very low resource treebanks consist of Buryat BDT (bxr), Kazakh KTB (kk), Kurmanji MG (kmr), Livvi KKPP (olo), and Upper Sorbian UFAL (hsb). The low resource set is made up of Belarusian HSE (be), Galician TreeGal (gl), Lithuanian HSE (lt), Marathi UFAL (mr), Old Russian RNC (orv), Tamil TTB (ta), and Welsh CCG (cy). We combined the training and development data (when available) to then split them 80|20. The statistics for the resulting splits are shown in Table 1. We use the original test data for analysis.

Artificial low resource data   We use Indonesian GSD (id), Irish IDT (ga), Japanese GSD (ja), and Wolof WTB (wo) to create artificially low resource treebanks. We take a sample of 100, 232, and 541 sentences from the training and development data. These are then split 80|20 for training and development data. We do this three times for each treebank size so we have multiple samples to verify our results. We use the original test data for analysis.

Augmented data   For the experiment using augmented data we use a subset of the smallest tree-banks, namely Kazakh, Kurmanji, and Upper Sorbian. We then generate data using the subtree swapping data augmentation technique of Dehouck and Gómez-Rodríguez (2020). We generate 10, 25, and 50 trees for each and we then split them 80|20. We do this three times for each number of generated trees. We use the original test data for analysis.

Subtree swapping   We gather all the sub-trees with a continuous span which has a NOUN, VERB, ADJ or PROPN as its root node. Other UPOS tags are not used due the likelihood of generating ungrammatical structures. With regards to the permitted relation of the root nodes, we consider all core arguments, all nominal dependents, and most non-core dependents (excluding discourse, expl and dislocated). Then given a tree, we swap one of its sub-trees with one from another tree given that their respective roots have the same UPOS tag, dependency relation and morphological features and given that the sub-trees are lexically different. We repeat the process a second time using a third tree. During this second swap, we do not

allow the previously swapped subtree to be altered again so as to avoid redundancy. For a more detailed description of this process see Dehouck and Gómez-Rodríguez (2020). We create all possible trees generated from the three original trees given the constraints described above, repeat this for each triplet of trees, and finally take a sample from this set of augmented data.

	Train		Dev	
	sents	tokens	sents	tokens
bxr	15	120	4	33
kk	24	395	7	134
kmr	16	192	4	50
olo	15	114	4	30
hsb	18	310	5	150
be	307	6,441	77	1,449
gl	480	12,317	120	3,119
lt	166	3,444	42	852
mr	335	2,751	84	686
orv	256	8,253	64	1903
ta	383	6,082	96	1,254
cy	491	10,719	123	2,616

Table 1: Number of trees in training and development splits as used for low resource UD treebanks.

Controlling UPOS accuracy   For each treebank size and split for the artificial low resource treebanks we trained taggers with varying accuracies (60, 66, 72, 78, 85, 89). We allowed a small window around the accuracy for each bin of ±0.25. Following a similar methodology to Anderson and Gómez-Rodríguez (2020) to obtain taggers with varying accuracies, we train the taggers as normal and save models when they reach a desired accuracy. We then train parsers using predicted tags from each of the taggers and use predicted tags at inference. For the data augmentation experiment we used accuracy bins of 41, 44, 48, and 51.

Network details   Both the taggers and parsers use word embeddings and character embeddings. The parsers use UPOS tag embeddings except for the MTL setup and the baseline models without tags. The embeddings are randomly initialised. The parsers consist of the embedding layer followed by BiLSTM layers and then a biaffine mechanism (Dozat and Manning, 2017). The taggers are similar but with an MLP following the BiLSTMs instead. We ran a hyperparameter search evaluated on the development data of Irish and Wolof. This resulted in 3 BiLSTM layers with 200 nodes, 100 dimensions for each embedding type with 100 dimensions for input to the character LSTM. The arc
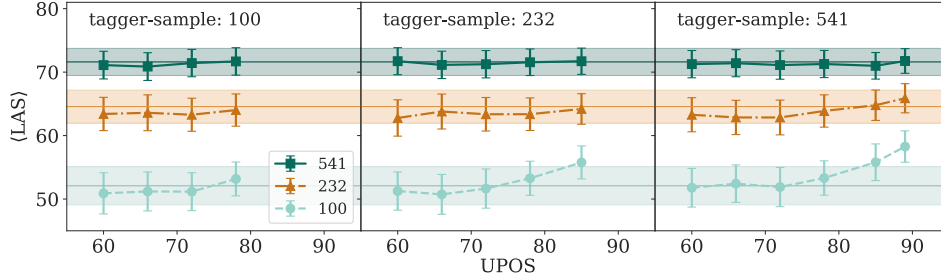
Figure 1: Impact of tagging accuracy for varying amounts of data for both taggers and parsers using artificial low resource data. The standard error of UPOS accuracy is not shown as it is very small ($< 0.1\%$ relative error for all bins). Horizontal lines and corresponding shaded area show the mean parsing performance and the standard error for the baseline parsers trained without UPOS tags.

MLP of the biaffine structure has 100 dimensions, whereas the relation MLP has 50.

3 Results and discussion

Table 2 shows the real low resource treebank results. Table 2a shows the results for the treebanks with less than 50 sentences. The performance is very low across the board so it is difficult to draw any substantial conclusions, however, using gold tags has a large impact over not using any, almost doubling the labeled attachment score. Also, using predicted tags does result in an increase on average, but Kazakh and Kurmanji lose almost a point. Further those two treebanks and also Buryat have reasonable gains when using the multi-task framework. The average multi-task score is strongly affected by the large drop seen for Upper Sorbian, which also suffers with respect to tagging accuracy when using the multi-task setup.

Table 2b shows the results for the low resource treebanks with less than 750 sentences. On average using predicted UPOS tags achieves a sizeable increase over not using any tags of about 1.2, despite the average tagging accuracy only being 85.89%. This suggests that in a lower resource setting the tagging accuracy doesn't have to be quite so high as is needed for high resource settings. Increases in performance are seen for all treebanks except Lithuanian and Tamil. While Lithuanian has the second lowest tagging score, Tamil has a fairly high score, so it seems that the accuracy needed is somewhat language-specific or at the very least data-dependent. The difference for the treebanks in Table 2b is almost 9 points higher for using gold tags. The multi-task performance is about 1.4 points less than using predicted tags on average. However, Lithuanian and Tamil obtain an increase in performance using the multi-task system in com-

parison to using predicted tags.

Figure 1 shows the average LAS performance for the parsers trained with the artificial low resource data. When the parsers have sufficient data, using UPOS tags doesn't offer any improvement in performance. For the parsers trained with 232 samples, there is a slight upward trend when using tags predicted from taggers trained with 541 samples. The improvement increases with respect to UPOS tag accuracy and exceeds the performance of the parsers trained with no UPOS tags. The most no-

| | UPOS | | LAS | | | |
	Single	Multi	None	Pred	Gold	Multi
bxr	48.72	48.34	10.45	12.36	20.31	14.41
kk	53.37	52.14	22.48	21.63	36.66	23.50
kmr	50.56	53.73	19.16	18.31	35.54	21.58
olo	37.84	37.37	9.74	10.89	17.54	7.59
hsb	53.44	47.28	18.36	20.03	41.88	14.66
avg	48.79	47.77	16.04	16.64	30.39	16.25

(a) Very low resource: less than 50 sentences.

| | UPOS | | LAS | | | |
	Single	Multi	None	Pred	Gold	Multi
be	92.82	87.29	61.82	64.91	68.87	62.28
gl	93.54	88.56	70.60	72.73	79.06	70.54
lt	79.25	71.51	37.17	35.94	48.30	38.96
mr	80.58	76.46	57.04	58.74	64.32	56.31
orv	87.77	81.60	49.53	51.34	60.24	50.33
ta	86.88	79.23	63.85	62.75	74.31	63.15
cy	91.77	86.41	72.10	72.93	80.71	73.00
avg	85.89	77.77	55.24	56.52	64.13	55.10

(b) Low resource: less than 750 sentences.

Table 2: Performance of different low resource parsers: using predicted UPOS tags as features (Pred), multi-task system where tagging is an auxiliary task to parsing (Multi), using gold UPOS tags as features (Gold), and without using UPOS tags as features (None). The accuracies of the predicted UPOS tags (Single) and that of the multi-task (Multi) are also reported.
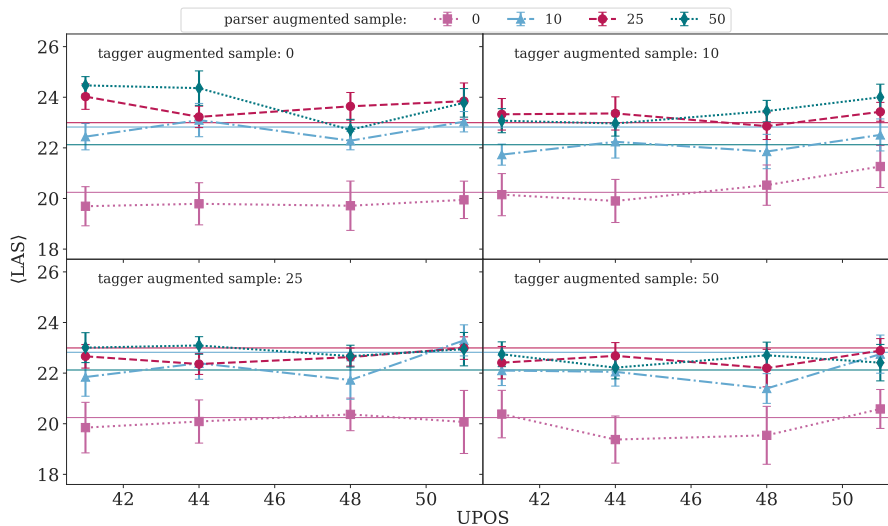
80

Figure 2: Impact of tagging accuracy for varying amounts of data for both taggers and parsers using augmented data (0, 10, 25, and 50 augmented trees) on top of the original gold data. The standard error of UPOS accuracy is not shown as it is very small ($< 0.1\%$ relative error for all bins). Horizontal lines show the mean parsing performance for the baseline parsers trained without UPOS tags (standard error not shown due to too much overlap between augmented data sample sizes).

ticeable improvement is for the parsers trained with only 100 samples. The impact of UPOS accuracy is clearer as the tagger sample size increases as higher accuracies can be obtained. The best performance is with the most accurate taggers (89%).

This is a potentially useful finding if annotators have little time, as annotating UPOS tags is much less time-sensitive and can help improve parsing performance if a limited number of tree-annotated sentences are available. However, taking parsers using only 100 fully-annotated training sentences as a baseline, the average performance using 232 parsed sentences without UPOS tags is over 10 points higher, whereas the increase gained training the taggers with 541 tagged sentences is only 5 points. So it is clear that if time permits such that annotators can increase the number of tree annotations, they will likely prove to be more useful. But UPOS tags could be obtained using projection methods and/or active learning techniques (Baldridge and Palmer, 2009; Das and Petrov, 2011; Garrette et al., 2013; Täckström et al., 2013). Also, multilingual projection methods could be used, but they typically generate trees as well as POS tags (Agić et al., 2016; Johannsen et al., 2016).

Figure 2 shows the impact of predicted UPOS accuracy when using data generated with subtree swapping augmentation. The first result worth noting is that the augmented data increases performance in this very low resource context. Across the board, the best performing parsers using aug-

mented data outperform the parsers trained only on gold data by 3-6 points which corroborates the findings in previous work. However, it appears that there is a limit to how much augmented data helps as the performance of the parsers which use 25 and 50 augmented instances is similar.

It also appears that this upper limit is even lower for training taggers with the best performance coming when using predicted tags from taggers utilising only 10 augmented samples or none at all. Using more invariably hurts performance no matter what accuracy the taggers obtained, as can be seen in the subplots showing the performance for parsers trained with predicted tags from taggers using 25 and 50 augmented samples. Also, there is no clear trend showing the impact of UPOS accuracy in this very low resource context.

4 Conclusion

We have presented results which suggest that lower accuracy taggers can still be beneficial when little data is available for training parsers, but this requires a high ratio of UPOS annotated data to tree annotated data. Experiments using artificial low resource treebanks highlight that this utility diminishes if the number of samples reaches a fairly small amount. We have also shown that very small treebanks can benefit from augmented data and utilise predicted UPOS tags even when they come from taggers with very low accuracy. Our experi-

81

ments haven't considered pretrained multilingual language models (LMs) which could potentially offset the small benefits of using POS tags. It would be interesting to develop this analysis further by testing whether the implicit information encoded in these LMs are more useful than explicit but potentially erroneous POS tag information. Finally, as one reviewer highlighted, the set of POS tags in the UD framework might just not be sufficiently informative in this setting. While this might be true, the greater contributing factor is surely the low accuracy of the taggers.

Acknowledgments

References

Željko Agić, Anders Johannsen, Barbara Plank, Héctor Martínez Alonso, Natalie Schluter, and Anders Søgaard. 2016. Multilingual projection for parsing truly low-resource languages. *Transactions of the Association for Computational Linguistics*, 4:301–312.

Mark Anderson and Carlos Gómez-Rodríguez. 2020. On the frailty of universal POS tags for neural UD parsers. In *Proceeding of the 24th SIGNLL Conference on Computational Natural Language Learning*, Online.

Jason Baldridge and Alexis Palmer. 2009. How well does active learning actually work? time-based evaluation of cost-reduction strategies for language documentation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 296–305.

Miguel Ballesteros, Chris Dyer, and Noah A Smith. 2015. Improved transition-based parsing by modeling characters instead of words with LSTMs. *arXiv preprint arXiv:1508.00657*.

Dipanjan Das and Slav Petrov. 2011. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 600–609.

Miryam de Lhoneux, Yan Shao, Ali Basirat, Eliyahu Kiperwasser, Sara Stymne, Yoav Goldberg, and Joakim Nivre. 2017. From raw text to universal dependencies-look, no tags! In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 207–217.

Mathieu Dehouck and Carlos Gómez-Rodríguez. 2020. Data augmentation via subtree swapping for dependency parsing of low-resource languages. In *to appear in the 28th International Conference on Computational Linguistics (COLING 2020)*, online.

Timothy Dozat and Christopher D Manning. 2017. Deep biaffine attention for neural dependency parsing. *Proceedings of the 5th International Conference on Learning Representations*.

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*, pages 20–30.

Kuzman Ganchev, Jennifer Gillenwater, and Ben Taskar. 2009. Dependency grammar induction via bitext projection constraints. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 369–377.

Dan Garrette, Jason Mielens, and Jason Baldridge. 2013. Real-world semi-supervised learning of pos-taggers for low-resource languages. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 583–592.

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Bootstrapping parsers via syntactic projection across parallel texts. *Natural Language Engineering*, 11(3):311–325.

Anders Johannsen, Željko Agić, and Anders Søgaard. 2016. Joint part-of-speech and dependency projection from multiple sources. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 561–566.

Katharina Kann, Ophélie Lacroix, and Anders Søgaard. 2020. Weakly supervised POS taggers perform poorly on truly low-resource languages. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(5):8066–8073.

Ryan McDonald, Slav Petrov, and Keith Hall. 2011. Multi-source transfer of delexicalized dependency parsers. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 62–72.

Aaron Smith, Miryam de Lhoneux, Sara Stymne, and Joakim Nivre. 2018. An investigation of the interactions between pre-trained word embeddings, character models and POS tags in dependency parsing. *arXiv preprint arXiv:1808.09060*.

Oscar Täckström, Dipanjan Das, Slav Petrov, Ryan McDonald, and Joakim Nivre. 2013. Token and type constraints for cross-lingual part-of-speech tagging. *Transactions of the Association for Computational Linguistics*, 1:1–12.

Jörg Tiedemann. 2015. Cross-lingual dependency parsing with universal dependencies and predicted pos labels. *Proceedings of the Third International Conference on Dependency Linguistics (Depling 2015)*, pages 340–349.

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)*, pages 1105–1116.

Daniel Zeman, Joakim Nivre, et al. 2020. Universal Dependencies 2.6. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Daniel Zeman and Philip Resnik. 2008. Cross-language parser adaptation between related languages. In *Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages*, pages 35–42.

Yu Zhang, Zhenghua Li, Houquan Zhou, and Min Zhang. 2020. Is POS tagging necessary or even helpful for neural dependency parsing? *arXiv preprint arXiv:2003.03204*.