We Need to Talk About train-dev-test Splits: Addendum

Rob van der Goot
IT University of Copenhagen
robv@itu.dk

1 Is a Tune-set More Effective in Low-resource scenarios?

In the experiments in the paper, we saw no significant difference in performance when using a tune split and when not using a tune split. One of the reviewers suggested to retry this experiment in a low-resource setup, another reviewer was wondering why we use parsing in these experiments. I was hoping to include experiments for this in the main paper, but I had some time and computational constraints that prevented this. I have now ran the experiments for the low resource setup, where we use a maximum number of 2,000 sentences for training, and present the results in this addendum.

Results can be found in Table 1. Again, none of the differences is significant. The number of different hyperparameters found is now a bit higher, but apparently none of them are important enough to lead to a large performance difference. This could also mean that the ranges of hyperparameters evaluated simply does not have a large impact on the performance. Perhaps even less training data, or other hyperparameters show a different effect. However, because the experiments are rather computationally expensive to run, and I do not want to keep trying new datasets until I find something that is significant (publication bias), I have not tried these experiments for other datasets.

It should again be noted that altough the ideal support for a tune split would be showing that using a tune split leads to improved performance, these results are also positive with respect to the tune split. As using a tune split leads to less overfitting on the test data, allows the dev data to be used for analysis, and at the same time it does not lead to decreased performance, it would still be beneficial to use it.

	MaChAmp			UUParser		
Dataset	Dif	-T	+T	Dif	-T	+T
grc_proiel	2/4	56.50	57.09	2/8	63.12	62.84
ar_padt	2/4	80.27	80.01	2/8	73.86	73.49
en_ewt	2/4	84.19	84.55	2/8	70.73	71.09
fi_tdt	3/4	79.09	79.38	0/8	65.70	66.25
zh_gsd	2/4	73.17	82.50	2/8	68.73	68.74
he_htb	1/4	82.62	82.72	1/8	69.09	69.07
ko_gsd	2/4	81.08	81.19	2/8	74.98	73.60
ru_gsd	2/4	83.63	88.32	0/8	79.39	79.27
sv_talbanken	2/4	81.68	81.93	2/8	68.06	68.51

Table 1: Results (LAS) of tuning with both strategies for a low-resource setup. Dif reports the number of optimal hyperparameters that differ between the two setups, -T(une) is using dev for model picking as well as hyperparameter-tuning, and +T(une) is our proposed setup.