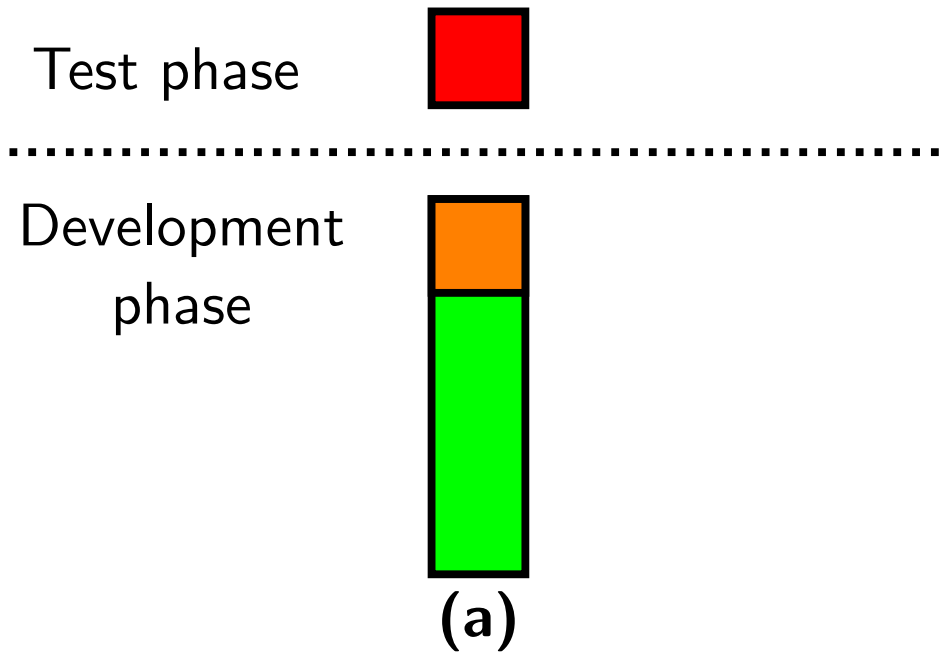
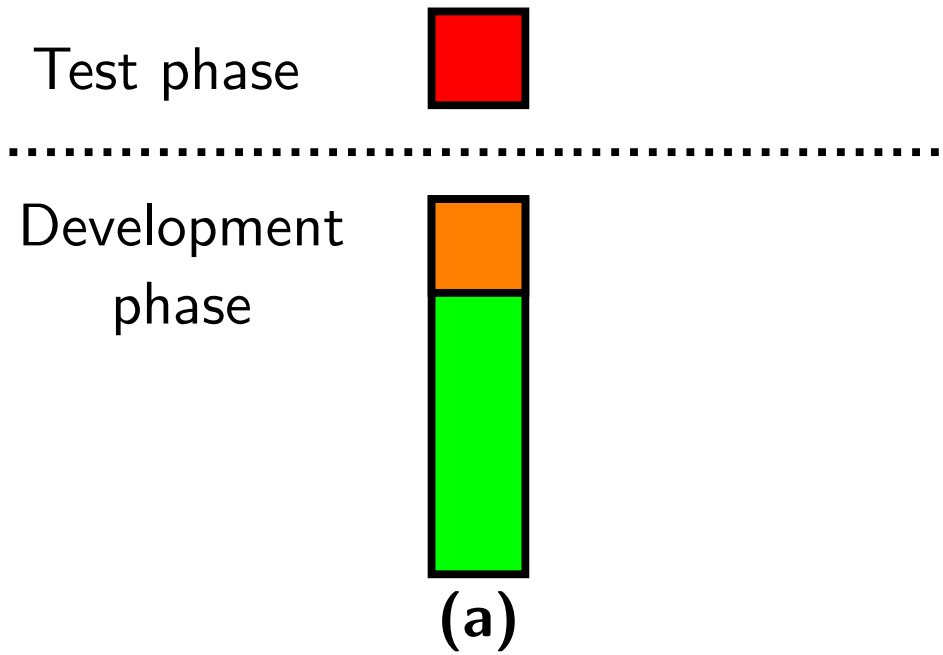
We Need to Talk About train-dev-test Splits

Rob van der Goot

Train-dev-test splits

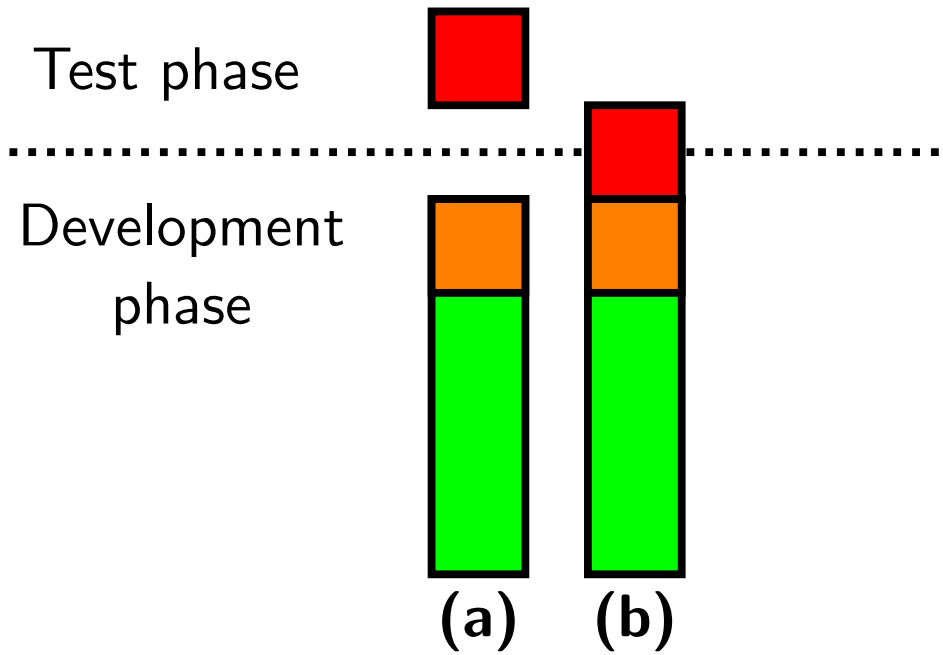


Train-dev-test splits

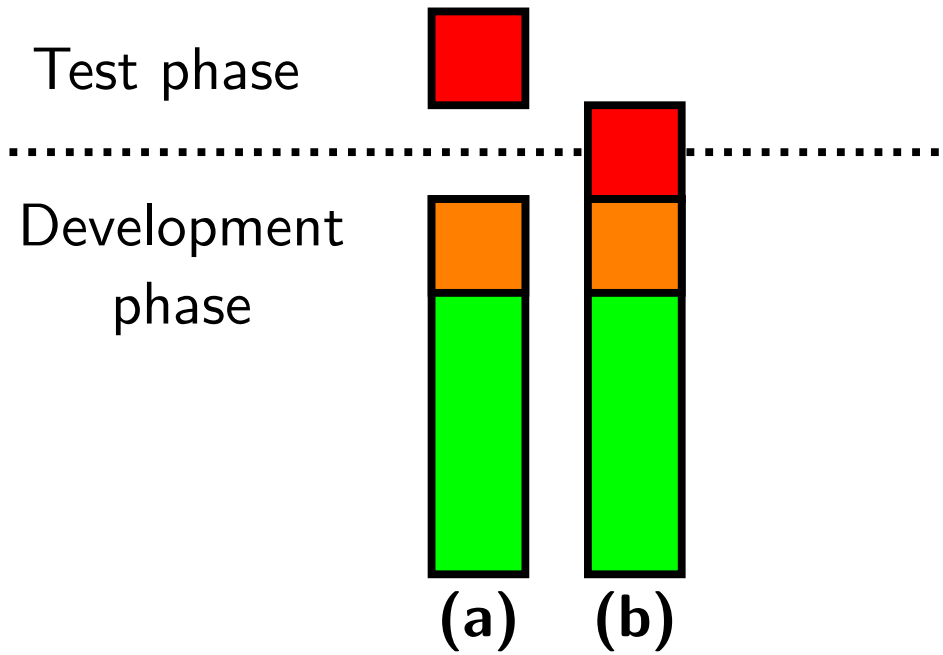


Neural networks $\mapsto$ Overfitting of design decisions on dev data!

Train-dev-test splits

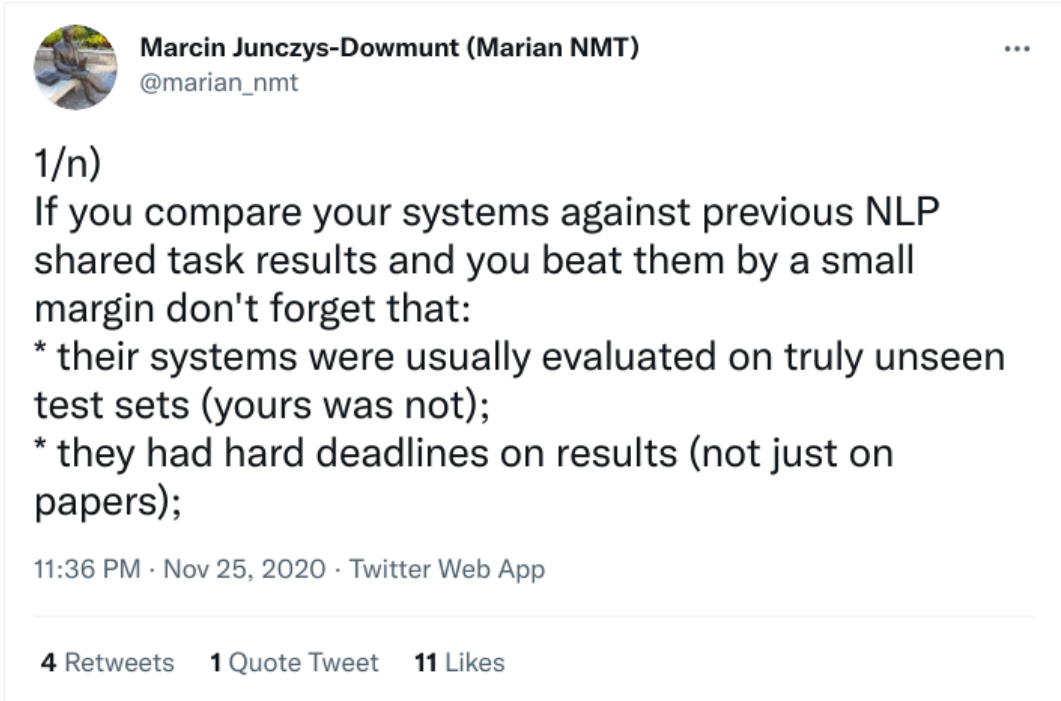


Train-dev-test splits



Neural networks $\mapsto$ Overfitting of design decisions on test data!

Use of test data



Marcin Junczys-Dowmunt (Marian NMT)
@marian_nmt

1/n)
If you compare your systems against previous NLP shared task results and you beat them by a small margin don't forget that:
* their systems were usually evaluated on truly unseen test sets (yours was not);
* they had hard deadlines on results (not just on papers);

11:36 PM · Nov 25, 2020 · Twitter Web App

4 Retweets **1** Quote Tweet **11** Likes

Use of test data

2021.emnlp.org/call-for-papers

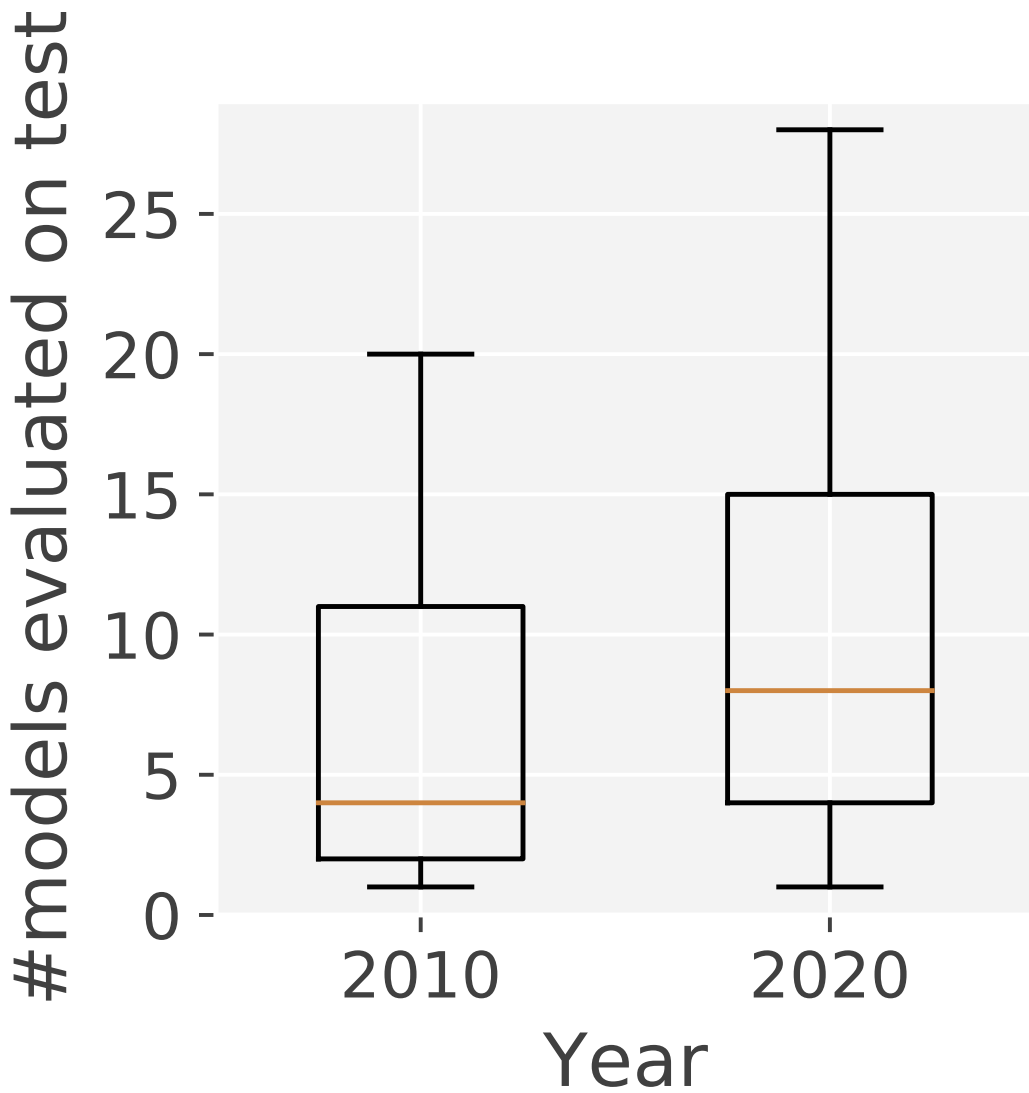
Reproducibility Criteria

During the submission process, authors will be asked to answer the questions from the Reproducibility Checklist. The checklist is intended as a reminder to help the authors improve reproducibility of their papers. The papers are not required to meet all reproducibility criteria listed. However, the answers will be made available to the reviewers. Reviewers will be asked to assess the reproducibility of the work as part of their reviews.

The following is a preliminary checklist we plan to use. For all reported experimental results:

- A clear description of the mathematical setting, algorithm, and/or model.
- Submission of a zip file containing source code, with specification of all dependencies, including external libraries, or a link to such resources (while still anonymized)
- Description of computing infrastructure used
- The average runtime for each model or algorithm (e.g., training, inference, etc.), or estimated energy cost
- Number of parameters in each model
- Corresponding validation performance for each reported test result
- Explanation of evaluation metrics used, with links to code

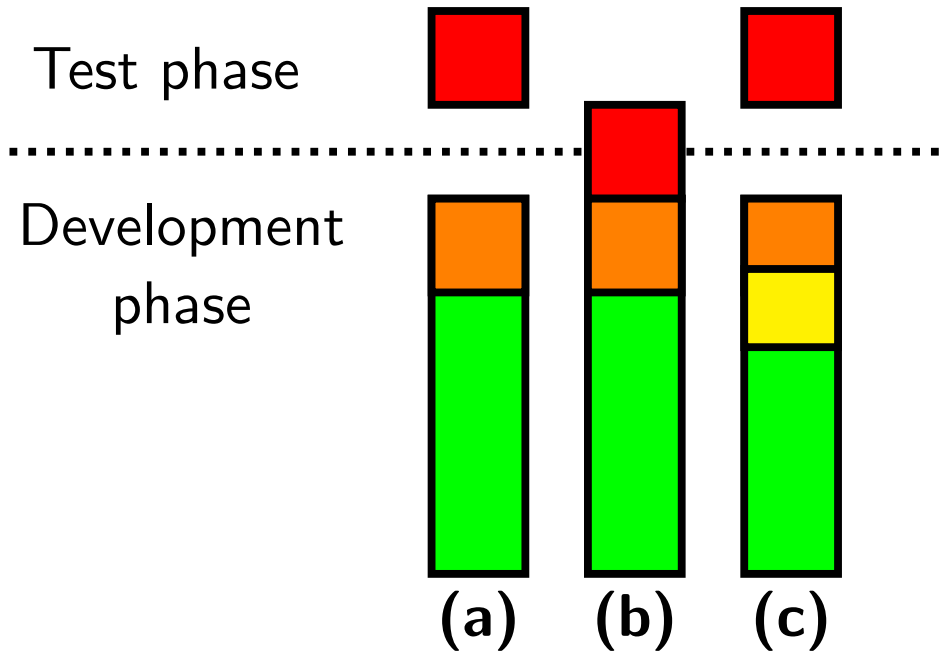
Use of test data



Problem

Now that the dev split is integrated into the training procedure, we have no datasplit left for comparing different versions of our models.

Solution: Tune split



But now we have less data for training!

But now we have less data for training!

- We can do development with train, tune and dev
- Get performance on test based on a model trained on train+tune and dev used for model picking

Is this new?

No:

- Cross-lingual learning with source language train+dev, target language dev+test
- Devtest set in machine translation

Is this new?

No:

- Cross-lingual learning with source language train+dev, target language dev+test
- Devtest set in machine translation
- In shared tasks the test split is enforced to be only used a few times
- Some online benchmarks keep the test set secret

Is this new?

No:

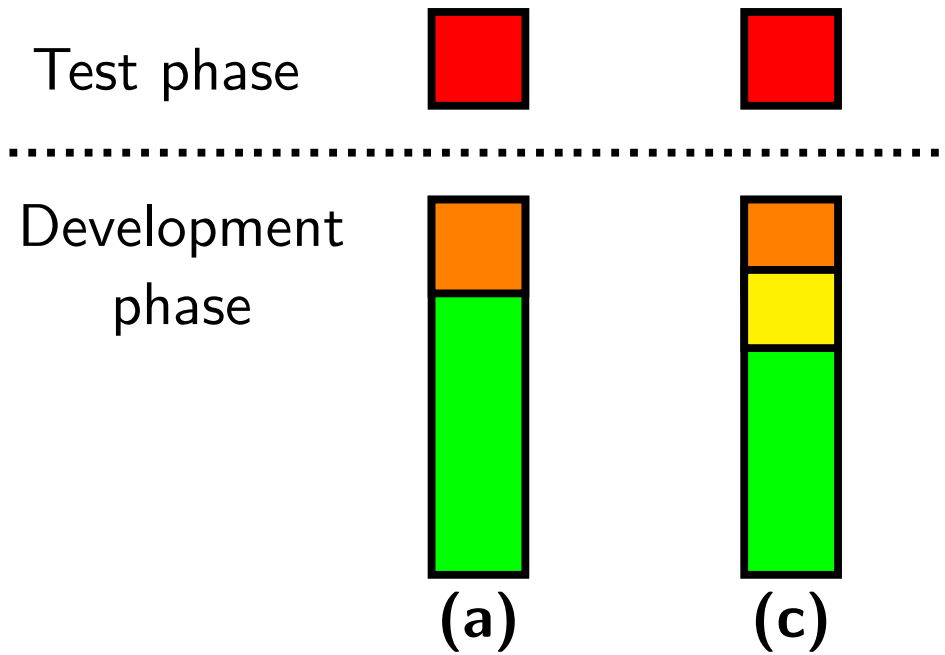
- Cross-lingual learning with source language train+dev, target language dev+test
- Devtest set in machine translation
- In shared tasks the test split is enforced to be only used a few times
- Some online benchmarks keep the test set secret
- Chen and Ritter (2020) propose other model-picking strategies

Experiments

- UD 2.8: subset from Smith et al. (2018b)
- Fine-tuning of MaChAmp and UUParser
- Compare using dev for model picking versus using tune for model picking

Experiments

- UD 2.8: subset from Smith et al. (2018b)
- Fine-tuning of MaChAmp and UUParser
- Compare using dev for model picking versus using tune for model picking



Experiments

- Concatenate train and dev
- Last 3,000 sentences used for 1,000 tune, dev and test
- Rest is training data

Experiments

A (better) alternative:

- $<$3,000 sentences: 1/4th tune, 1/4th dev, 2/4th train
- $>$3,000 sentences: 750 tune, 750 dev, rest train

Experiments

Dataset	MaChAmp			UUParser		
	Dif	-T	+T	Dif	-T	+T
grc_proiel	2/4	72.28	72.19	2/7	78.17	77.38
ar_padt	1/4	82.11	81.82	0/7	77.60	77.63
en_ewt	1/4	88.89	88.90	1/7	82.64	82.90
fi_tdt	2/4	88.41	87.85	1/7	80.50	80.81
zh_gsd	1/4	83.13	82.66	0/7	69.67	69.27
he_htb	2/4	84.49	84.33	1/7	73.22	73.30
ko_gsd	2/4	81.99	82.32	0/7	77.28	77.15
ru_gsd	2/4	88.51	88.48	1/7	80.14	79.84
sv_talbanken	1/4	82.76	82.89	1/7	71.11	71.40

Table: Results (LAS) of tuning with both strategies. 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. *Statistical significant.

Experiments

	MaChAmp			UUParser		
Dataset	Dif	-T	+T	Dif	-T	+T
grc_proiel	2/4	72.28	72.19	2/7	78.17	77.38
ar_padt	1/4	82.11	81.82	0/7	77.60	77.63
en_ewt	1/4	88.89	88.90	1/7	82.64	82.90
fi_tdt	2/4	88.41	87.85	1/7	80.50	80.81
zh_gsd	1/4	83.13	82.66	0/7	69.67	69.27
he_htb	2/4	84.49	84.33	1/7	73.22	73.30
ko_gsd	2/4	81.99	82.32	0/7	77.28	77.15
ru_gsd	2/4	88.51	88.48	1/7	80.14	79.84
sv_talbanken	1/4	82.76	82.89	1/7	71.11	71.40

Table: Results (LAS) of tuning with both strategies. 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. *Statistical significant.

Experiments

Dataset	MaChAmp			UUParser		
	Dif	-T	+T	Dif	-T	+T
grc_proiel	2/4	72.28	72.19	2/7	78.17	77.38
ar_padt	1/4	82.11	81.82	0/7	77.60	77.63
en_ewt	1/4	88.89	88.90	1/7	82.64	82.90
fi_tdt	2/4	88.41	87.85	1/7	80.50	80.81
zh_gsd	1/4	83.13	82.66	0/7	69.67	69.27
he_htb	2/4	84.49	84.33	1/7	73.22	73.30
ko_gsd	2/4	81.99	82.32	0/7	77.28	77.15
ru_gsd	2/4	88.51	88.48	1/7	80.14	79.84
sv_talbanken	1/4	82.76	82.89	1/7	71.11	71.40

Table: Results (LAS) of tuning with both strategies. 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. *Statistical significant.

Experiments

Dataset	MaChAmp			UUParser		
	Dif	-T	+T	Dif	-T	+T
grc_proiel	2/4	72.28	72.19	2/7	78.17	77.38
ar_padt	1/4	82.11	81.82	0/7	77.60	77.63
en_ewt	1/4	88.89	88.90	1/7	82.64	82.90
fi_tdt	2/4	88.41	87.85	1/7	80.50	80.81
zh_gsd	1/4	83.13	82.66	0/7	69.67	69.27
he_htb	2/4	84.49	84.33	1/7	73.22	73.30
ko_gsd	2/4	81.99	82.32	0/7	77.28	77.15
ru_gsd	2/4	88.51	88.48	1/7	80.14	79.84
sv_talbanken	1/4	82.76	82.89	1/7	71.11	71.40

Table: Results (LAS) of tuning with both strategies. 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. *Statistical significant.

Thanks for your attention!

- Source code and "split code" available:
  `http://bitbucket.org/robvanderg/tuneset`