Why Comparing Single Performance Scores Does Not Allow to Draw Conclusions About Machine Learning Approaches

Nils Reimers and **Iryna Gurevych**
Ubiquitous Knowledge Processing Lab (UKP) and Research Training Group AIPHES
Department of Computer Science, Technische Universität Darmstadt
www.ukp.tu-darmstadt.de

Abstract

Developing *state-of-the-art approaches* for specific tasks is a major driving force in our research community. Depending on the prestige of the task, publishing it can come along with a lot of visibility. The question arises how reliable are our evaluation methodologies to compare approaches?

One common methodology to identify the state-of-the-art is to partition data into a train, a development and a test set. Researchers can train and tune their approach on some part of the dataset and then select the model that worked best on the development set for a final evaluation on unseen test data. Test scores from different approaches are compared, and performance differences are tested for statistical significance.

In this publication, we show that there is a high risk that a statistical significance in this type of evaluation is not due to a superior learning approach. Instead, there is a high risk that the difference is due to chance. For example for the CoNLL 2003 NER dataset we observed in up to 26% of the cases type I errors (false positives) with a threshold of $p < 0.05$, i.e., falsely concluding a statistically significant difference between two identical approaches.

We prove that this evaluation setup is unsuitable to compare learning approaches. We formalize alternative evaluation setups based on score distributions.

1 Introduction

Given two machine learning approaches, approach **A** and approach **B**, for a certain dataset, how can we decide which approach is more accurate for this task? This is a fundamental question in our community, where a lot of effort is spent to identify new *state-of-the-art* approaches. Hence, we want that the evaluation setup is not impacted by random chance and we should draw the same conclusion if the experiment is reproduced.

While different evaluation setups exist, one fairly common evaluation setup is to partition annotated data into a training, development and test set. Approaches are trained and tuned on the train and development set, and then a performance score on a held-out test set is computed. The approach with the higher test performance score is observed as superior[1].

As the test set is a finite sample, the test score differs from the (hypothetical) performance on the complete data distribution. A significance test on the test set is used to reduce the risk that chance induced from the finite test sample is the explanation for the difference. If the difference is significant, it is usually accepted that one approach is superior to the other.

This evaluation methodology is often used in scientific publications and for shared tasks in our field, for example, it is commonly used for the shared tasks at the International Workshop on Semantic Evaluation (SemEval) and for the shared tasks from the Conference on Computational Natural Language Learning (CoNLL). The participants either submit the output of their system for the unlabeled test data to the task organizers or, as it was the case for the CoNLL 2017 shared task on multilingual parsing (Zeman et al., 2017), participants submitted their sys-

[1] In this paper, we only judge approaches based on how accurate those are given a specific performance measure. For real-world applications, superiority can mean many distinct things that are not related to accuracy.

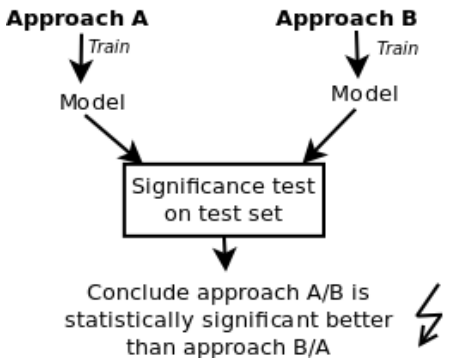

Figure 1: Common evaluation methodology to compare two approaches for a specific task.

tem to a cloud-based evaluation platform where it was applied to new data. To identify if differences are significant, the organizers used paired bootstrap resampling. Hiding the test data from the participants eliminates the risk that information about the test data is used for the design of the approach. Depending on the prestige of the shared task, winning it can come along with a lot of visibility. The winning approach is often part of future research or serves as a baseline for new approaches.

The question arises how reliable is this evaluation setup and how reliable are shared tasks to identify the best approach? As our results show, this evaluation methodology is incapable to distinguish which learning approach is superior for the studied task.

In this paper, we show that there is a high risk that chance, and not a superior design, leads to significant differences. For example, for the CoNLL 2003 shared task on NER, we compared two identical neural networks with each other. In 22% of the cases, we observed a significant difference in test score with $p < 0.05$. By implication, if we observe a significant difference in test performance, we cannot be certain if the difference is due to a superior approach or due to luck. The issue is not a flawed significance test but lies in wrongly drawn conclusions.

In the context of this paper, it is important to notice the difference between *models* and *learning approach*. A *learning approach* describes the holistic setup to solve a certain optimization problem. For neural networks, this would be the network architecture, the optimization algorithm, the loss-function etc. A *model* is a specific configuration of the weights for this architecture.

A significance test for a specific model can only check if the model will likely perform better for the whole data distribution. However, we often observe that the conclusion is drawn that a superior model implies a superior approach for that task. For example, for the shared task SemEval-2017 on semantic textual similarity (STS) the task organizers conclude that the model from the winning team is *"the best overall system"* (Cer et al., 2017). Szegedy et al. (2015) conclude that the winning model from Clarifai for the ImageNet 2013 challenge was the *"year's best approach"*.

The contribution in this paper is to show, that this conclusion cannot be drawn for *non-deterministic learning approaches*[2], like neural networks. Generating a model with superior (test) performance does not allow the conclusion that the learning approach is superior for that task and data split. If two similar approaches are compared, then there is a high risk that a luckier sequence of random numbers, and not the architecture, decides which approach generates a significantly better test performance.

We argue for a change in the evaluation paradigm of machine learning systems. Instead of comparing and reporting individual system runs, we propose training approaches multiple times and comparing score distributions (section 7).

2 Related Work

No evaluation setup is perfect and many points are discussable, for example, the right evaluation metric, how to aggregate results, and many more points (Japkowicz and Shah, 2011). With a different evaluation setup, we might draw different conclusions. However, to allow a comparison of approaches, the community often uses common evaluation setups. In a lot of cases, these evaluation setups were established in shared tasks and are used long after the shared task. For example, the dataset and the setup of

[2]We define a learning approach as non-deterministic if it uses a sequence of random numbers to solve the optimization problem. Our observations are extendable to deterministic approaches that have tunable hyperparameters.

the CoNLL 2003 shared task on NER are still widely used to evaluate new approaches to detect named entities.

One commonly used methodology to compare machine learning approaches is described by Bishop (2006) (p. 32): "*If data is plentiful, then one approach is simply to use some of the available data to train a range of models, [...], and then to compare them on independent data, sometimes called a validation set, and select the one having the best predictive performance. [...] it may be necessary to keep aside a third test set on which the performance of the selected model is finally evaluated.*"

In order to make contributions by different researchers comparable, a popular tool is to use common dataset. Well known examples are the CoNLL 2003 dataset for NER or the CoNLL 2009 dataset for parsing. For those tasks and datasets, new approaches are trained on the provided data, and the test score is compared against published results.

As the test set is finite in size, there is a chance that a model achieves a better score on the test set, but would not yield a better score on the data population as a whole. To guard against this case, a significance test like the approximate randomized test (Riezler and Maxwell, 2005) or the bootstrap test (Berg-Kirkpatrick et al., 2012) can be applied. Those methods test the null hypothesis that both models would perform equally on the population as a whole. Significance tests typically estimate the confidence p, which should be an upper-bound for the probability of a type I error (a false positive error).

Training non-deterministic approaches a single time and comparing test scores can be misleading. It is known that, for example, neural networks converge to different points depending on the sequence of random numbers. However, not all convergence points generalize equally well to unseen data (Hochreiter and Schmidhuber, 1997; LeCun et al., 1998; Erhan et al., 2010). In our previous publication (Reimers and Gurevych, 2017b), we showed for the BiLSTM-CRF architecture for NLP sequence tagging tasks that the performance can vary depending on the random seed value. For the system by Ma and Hovy (2016) we showed that the F_1-score on the CoNLL 2003 NER dataset can vary between 89.88% and 91.00% and for the system by Lample et al. (2016) that the performance can vary between 90.19% and 90.81% depending on the random seed value. For some random seed values, the network converged to a poor minimum that generalizes badly on unseen data.

However, we are often only interested in the best performance an approach can achieve, for example, after tuning the approach. Failed attempts, like a random initialization that converged to a poor minimum, are often ignored. We eliminate these failed attempts by evaluating the models on a development set. For the final evaluation, we select only the model that performed best on the development set. The question arises if this is a valid evaluation methodology to compare learning approaches for a task?

To our knowledge, this has not been studied before. In section 3.2 we formalize this type of evaluation. In section 5.2 we show empirically for seven NLP sequence tagging tasks that this evaluation method is incapable to compare learning approaches. We then present a proof in section 6 that this evaluation method is in general incapable to compare learning approaches for any tasks, learning approach, and statistical significance test.

3 Evaluation Methodologies based on Single Scores

This section formalizes evaluation methods that are based on single model comparisons. Note, in all cases we assume a fixed train, development, and test set for example from a shared task.

3.1 Single Run Comparison

The first evaluation method is to train both approaches a single time and to compare the test scores.

Evaluation 1. Given two approaches, we train both approaches a single time to generate the models A_i and B_j. We define $\Psi_{A_i}^{(Test)}$ as the test score for model A_i and $\Psi_{B_j}^{(Test)}$ as the test score for model B_j. We call approach A is superior over approach B if and only if $\Psi_{A_1}^{(test)} > \Psi_{B_1}^{(test)}$ and the difference is statistical

significant. Commonly used significance tests are an approximate randomized test or a bootstrap test (Riezler and Maxwell, 2005).

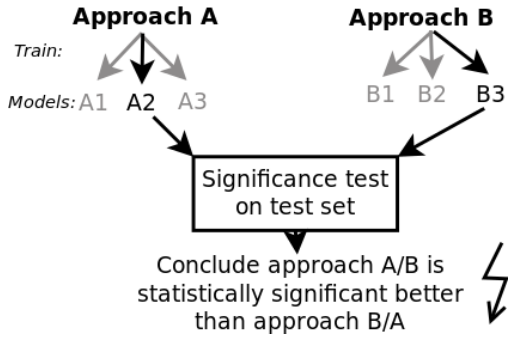

Figure 2: Single score comparison for non-deterministic learning approaches (Evaluation 1).

Non-deterministic learning approaches, like neural networks, can produce many distinct models $A_1, ..., A_n$. Which model will be produced depends on the sequence of random numbers and cannot be determined in advance.

Figure 2 illustrates the issue of this evaluation methodology for non-deterministic learning approaches. Approach A produces the model A_2, while approach B the model B_3. Model A_2 might be significantly better than B_3, however, it might be worse than the other models B_1 or B_2.

3.2 Best Run Comparison

For shared tasks, the participants are not restricted to train their approach only once. Instead, they can train multiple models and can tune the parameters on the development set. For the final evaluation, they usually must select one model that is compared to the submissions from other participants. A similar process can often be found in scientific publications, where authors tune the approach on a development set and report the test score from the model that performed best on the development set. This form of evaluation is formalized in the following (depicted in Figure 3).

Evaluation 2. Given two approaches and we sample from each multiple models. Approach A produces the models $A_1, ..., A_n$ and approach B the models $B_1, ..., B_m$ with sufficiently large numbers of n and m. We define A_* as the best model from approach A and B_* as the best model from approach B. Bishop (2006) defines the best model as the model that performed best on the unseen development set:

$$A_* = \text{argmax}_{A_i \in \{A_1, ... A_n\}}(\Psi^{(dev)}_{A_i})$$
$$B_* = \text{argmax}_{B_i \in \{B_1, ... B_m\}}(\Psi^{(dev)}_{B_i})$$

With $\Psi^{(dev)}$ the performance score on the development set. We call approach A is superior over approach B iff $\Psi^{(test)}_{A_*} > \Psi^{(test)}_{B_*}$ and the difference is significant.

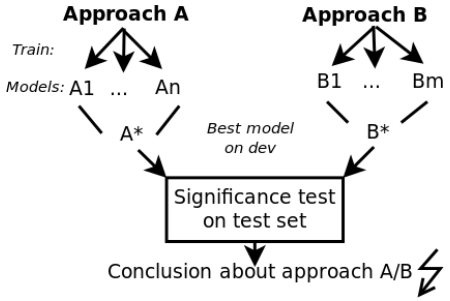

Figure 3: Illustration of model tuning and comparing the best models A_* and B_* (Evaluation 2).

The main contribution in this paper is to show that the conclusion $\Psi^{(test)}_{A_*} > \Psi^{(test)}_{B_*} \Rightarrow$ *approach A better than approach B* is wrong. This implies that this evaluation methodology is unsuitable for shared tasks and research publications.

4 Experimental Setup

We demonstrate that Evaluation 1 and Evaluation 2 fail to identify that two learning approaches are the same. By implication, a significant difference in test score does not allow the conclusion that one approach is better than the other.

We compare a learning approach A against itself, which we call approach A and $\tilde{A}$ hereafter. Approach A and $\tilde{A}$ use the same code, with the same configuration and are executed on the same computer. The only difference is that the sequence of random number changes each time.

A suitable evaluation method should conclude that there is no significant difference between A and $\tilde{A}$

in most cases. We use $p = 0.05$ as a threshold, hence, we would expect that a significant difference between A and $\tilde{A}$ only occurs in at most 5% of the cases.

4.1 Datasets

As benchmark tasks, we use seven common NLP sequence tagging tasks. We use the CoNLL 2000 dataset for Chunking, the CoNLL 2003 NER dataset for Named Entity Recognition for English and for German, the ACE 2005 dataset with the split by Li et al. (2013) for entity and event detection, the TempEval 3 event detection dataset[3], and the GermEval 2014 dataset for NER in German. We evaluate all tasks in terms of F_1-score.

4.2 Network Architecture

We use the BiLSTM-CRF architecture we described in (Reimers and Gurevych, 2017a).[4] We use 2 hidden layers, 100 hidden units each, variational dropout (Gal and Ghahramani, 2016) of 0.25 applied to both dimensions, Nadam as optimizer (Dozat, 2015), and a mini-batch size of 32 sentences. For the English datasets, we use the pre-trained embeddings by Komninos and Manandhar (2016). For the German datasets we used the embeddings by Reimers et al. (2014).

4.3 Training

In total, we trained 100,000 models for each task with different random seed values. We randomly assign 50,000 models to approach A while the other models are assigned to approach $\tilde{A}$.

For simplification, we write those models as two matrices with 50 columns and 1,000 rows each:

$$[A_i^{(j)}] \qquad [\tilde{A}_i^{(j)}]$$

with $i = 1, \ldots, 50$ and $j = 1, \ldots, 1000$. Each model $A_i^{(j)}$ has a development score $\Psi_{A_i^{(j)}}^{(dev)}$ and test score $\Psi_{A_i^{(j)}}^{(test)}$.

Model $A_*^{(j)}$ marks the model with the highest development score from the row $A_{1 \leq i \leq 50}^{(j)}$ and $\tilde{A}_*^{(j)}$ is the model with the highest development score from $\tilde{A}_{1 \leq i \leq 50}^{(j)}$. Hence, we test Evaluation 2 with $n = m = 50$.

4.4 Statistical Significance Test

We use the bootstrap method by Berg-Kirkpatrick et al. (2012) with 10,000 samples to test for statistical significance between test performances with a threshold of $p < 0.05$. We also tested the approximate randomized test, and the results were similar.

For Evaluation 1, we test on statistical significance between the models $A_i^{(j)}$ and $\tilde{A}_i^{(j)}$ for all i and j. For Evaluation 2, we test on statistical significance between $A_*^{(j)}$ and $\tilde{A}_*^{(j)}$ for $j = 1, \ldots, 1000$.

5 Results

We compute in how many cases the bootstrap method finds a statistically significant difference. Further, we compute the average F_1 test-score difference τ for pairs with an estimated p-value between 0.04 and 0.05. This value can be seen as a threshold: If the F_1-score difference is larger than this threshold, there is a high chance that the bootstrap method testifies a statistical significance between the two models.

Further, we compute the differences between the test performances for approach A and $\tilde{A}$. For Evaluation 1, we compute $\Delta^{(test),(i,j)} = |\Psi_{A_i^{(j)}}^{(test)} - \Psi_{\tilde{A}_i^{(j)}}^{(test)}|$. For Evaluation 2, we compute:

$$\Delta^{(test),(j)} = |\Psi_{A_*^{(j)}}^{(test)} - \Psi_{\tilde{A}_*^{(j)}}^{(test)}|.$$

For those delta values we compute a 95% percentile $\Delta_{95}^{(test)}$. The value indicates that a difference in the test score for a given task should be higher than $\Delta_{95}^{(test)}$, otherwise there is a chance greater 5% that the difference is due to chance for the given task and the given network architecture.[5]

[3] We used a random fraction of the documents in the training set to form a development set with approximately the size of the test set.

[4] https://github.com/UKPLab/emnlp2017-bilstm-cnn-crf

[5] Note that $\Delta_{95}^{(test)}$ depends on the used machine learning approach and the specific task.

Task	Threshold τ	% significant	$\Delta_{95}^{(test)}$	$\Delta_{Max}^{(test)}$
ACE 2005 - Entities	0.65	28.96%	1.21	2.53
ACE 2005 - Events	1.97	34.48%	4.32	9.04
CoNLL 2000 - Chunking	0.20	18.36%	0.30	0.56
CoNLL 2003 - NER-En	0.42	31.02%	0.83	1.69
CoNLL 2003 - NER-De	0.78	33.20%	1.61	3.36
GermEval 2014 - NER-De	0.60	26.80%	1.12	2.38
TempEval 3 - Events	1.19	10.72%	1.48	2.99

Table 1: The same BiLSTM-CRF approach was evaluated twice under Evaluation 1. The threshold column depicts the average difference in percentage points F_1-score for statistical significance with $0.04 < p < 0.05$. The *% significant* column depicts the ratio how often the difference between $A_i^{(j)}$ and $\tilde{A}_i^{(j)}$ is significant. Δ_{95} depicts the 95% percentile of differences between $A_i^{(j)}$ and $\tilde{A}_i^{(j)}$. $\Delta_{Max}^{(test)}$ shows the largest difference.

5.1 Comparing Single Performance Scores

Table 1 depicts the main results for Evaluation 1. For the *ACE 2005 - Events* task, we observe in 34.48% of the cases a significant difference between the models $A_i^{(j)}$ and $\tilde{A}_i^{(j)}$. For the other tasks, we observe similar results and between 10.72% and 33.20% of the cases are statistically significant.

The average F_1-score difference for statistical significance for the *ACE 2005 - Events* task is $\tau = 1.97$ percentage points. However, we observe that the difference between $A_i^{(j)}$ and $\tilde{A}_i^{(j)}$ can be as large as 9.04 percentage points F_1. While this is a rare outlier, we observe that the 95% percentile $\Delta_{95}^{(test)}$ is more than twice as large as τ for this task and dataset.

We conclude that training two non-deterministic approaches a single time and comparing their test performances is insufficient if we are interested to find out which approach is superior for that task.

5.2 Selecting the Best out of n Runs

Non-deterministic approaches can produce weak as well as strong models as shown in the previous section. Instead of training those a single time, we tune the approach and only compare the "best" model for each approach, i.e., the models that performed best on the development set. This evaluation method was formalized in Evaluation 2.

Table 2 depicts the results of this experiment. For all tasks, we observe small Spearman's rank correlation ρ between the development and the test score. The low correlation indicates that a run with high development score doesn't have to yield a high test score.

For the *ACE 2005 - Events* task, we observe a significant difference between $A_*^{(j)}$ and $\tilde{A}_*^{(j)}$ in 29.08% of the cases. We observe for this task that the difference in test score can be as large as 7.98 percentage points F_1-score between $A_*^{(j)}$ and $\tilde{A}_*^{(j)}$.

As before, we observe that $\Delta_{95}^{(test)}$ is much larger than τ, i.e. test performances of A_* vary to a large degree, larger than the threshold τ for statistical significance.

The table also depicts $\Delta_{95}^{(dev)}$, the 95% percentile of differences in terms of development performance. We observe a large discrepancy between $\Delta_{95}^{(dev)}$ and $\Delta_{95}^{(test)}$: For the 1,000 rows, we were able to find models $A_*^{(j)}$ and $\tilde{A}_*^{(j)}$ that performed comparably on the development set. However, their performance differs largely on the actual test set.

We studied if the value of statistically significant differences between A_* and $\tilde{A}_*$ depends on n, the number of sampled models. Figure 4 depicts the ratio for different n-values for the CoNLL 2003 NER task in English. We observe that the ratio of significant differences decreases with increasing number of sampled models n. However, the ratio stays flat after about 40 to 50 sampled models. For $n = 100$ we observe that 21.06% of the pairs are significant different with a $p < 0.05$ value.

Task	Spearman ρ	Threshold τ	% significant	$\Delta_{95}^{(dev)}$	$\Delta_{95}^{(test)}$	$\Delta_{Max}^{(test)}$
ACE 2005 - Entities	0.153	0.65	24.86%	0.42	1.04	1.66
ACE 2005 - Events	0.241	1.97	29.08%	1.29	3.73	7.98
CoNLL 2000 - Chunking	0.262	0.20	15.84%	0.10	0.29	0.49
CoNLL 2003 - NER-En	0.234	0.42	21.72%	0.27	0.67	1.12
CoNLL 2003 - NER-De	0.422	0.78	25.68%	0.58	1.44	2.22
GermEval 2014 - NER-De	0.333	0.60	16.72%	0.48	0.90	1.63
TempEval 3 - Events	-0.017	1.19	9.38%	0.74	1.41	2.57

Table 2: The same BiLSTM-CRF approach was evaluated twice under Evaluation 2. The threshold column depicts the average difference in percentage points F_1-score for statistical significance with $0.04 < p < 0.05$. The *% significant* column depicts the ratio how often the difference between $A_*^{(j)}$ and $\tilde{A}_*^{(j)}$ is significant. Δ_{95} depicts the 95% percentile of differences between $A_*^{(j)}$ and $\tilde{A}_*^{(j)}$. $\Delta_{Max}^{(test)}$ shows the largest difference.

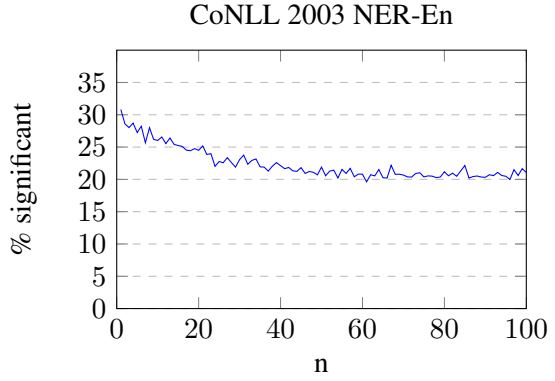

Figure 4: Ratio of statistically significant differences between A_* and $\tilde{A}_*$ for different n-values.

6 Why Comparing Best Model Performances is Insufficient

While it is straightforward to understand why Evaluation 1 is improper for non-deterministic machine learning approaches, it is less obvious why this is also the case for Evaluation 2. If we ignore the bad models, where the approach did not converge to a good performance, why can't we evaluate the best achievable performances of approaches?

The issue is not the significance test but has to do with the wrong conclusions we draw from a significant difference. The null-hypothesis for, e.g., the bootstrap test is that two compared models would perform not differently on the complete data distribution. However, it is wrong to conclude from this that one approach is capable of producing better models than the other approach. The issue is that selecting a model with high test / true performance is only possible to a certain degree and the uncertainty depends on the development set.

We write the (hypothetical) performance on the complete data distribution as $\Psi^{(true)}$. The development and test score are finite approximations of this true performance of a model.

We can rewrite the development score as $\Psi^{(dev)} = \Psi^{(true)} + \mathcal{X}^{(dev)}$ and the test score as $\Psi^{(test)} = \Psi^{(true)} + \mathcal{X}^{(test)}$. $\mathcal{X}^{(dev)}$ and $\mathcal{X}^{(test)}$ are two random variables with unknown means and variances stemming from the finite sizes of development and test set.

Given two models A_* and B_*, the significance test checks the null hypothesis whether $\Psi_{A_*}^{(true)}$ is equal to $\Psi_{B_*}^{(true)}$ given the two results on the test set.

When we select the models A_* and B_* based on their performance on the development set, we face the issue that the true performance is not monotone in the development score.

Assume we have models A_1 and A_2 with identical development performance. The development performance might be:

$$\Psi_{A_1}^{(dev)} = \Psi_{A_1}^{(true)} + \mathcal{X}_{A_1}^{(dev)} = 80\% - 2\% = 78\%$$
$$\Psi_{A_2}^{(dev)} = \Psi_{A_2}^{(true)} + \mathcal{X}_{A_2}^{(dev)} = 76\% + 2\% = 78\%$$

The test performances might be:

$$\Psi_{A_1}^{(test)} = 80\% + 1\% = 81\%$$
$$\Psi_{A_2}^{(test)} = 76\% - 1\% = 75\%$$

We compare this against model B_* from approach B, which as a test performance of $\Psi_{B_*}^{(test)} = 79\%$:

If we select A_1 for the comparison against B_*, the significance test might correctly identify that A_1 has a significantly lower test performance than B_*. However, if we select model A_2, the significance test might identify that B_* has a significantly higher test performance than A_2. As we do not know which model, A_1 or A_2, to select for Evaluation 2, the outcome of Evaluation 2 is up to chance. If we select A_1, we might conclude that approach A is better than approach B, if we select A_2, we might conclude the opposite.

In summary, a significance test based on a single model performance can only identify which model is better but does not allow the conclusion which learning approach is superior.

6.1 Distribution of $\Psi_{A_1}^{(test)} - \Psi_{A_2}^{(test)}$

We are interested to which degree the test score can vary for two models with identical development scores.

We can write the scores as:

$$\Psi_{A_1}^{(dev)} = \Psi_{A_1}^{(true)} + \mathcal{X}_{A_1}^{(dev)}$$
$$\Psi_{A_2}^{(dev)} = \Psi_{A_2}^{(true)} + \mathcal{X}_{A_2}^{(dev)}$$
$$\Psi_{A_1}^{(test)} = \Psi_{A_1}^{(true)} + \mathcal{X}_{A_1}^{(test)}$$
$$\Psi_{A_2}^{(test)} = \Psi_{A_2}^{(true)} + \mathcal{X}_{A_2}^{(test)}$$

We assume $\Psi_{A_1}^{(dev)} = \Psi_{A_2}^{(dev)}$, hence:

$$\Psi_{A_1}^{(true)} + \mathcal{X}_{A_1}^{(dev)} = \Psi_{A_2}^{(true)} + \mathcal{X}_{A_2}^{(dev)}$$
$$\Rightarrow \Psi_{A_1}^{(true)} - \Psi_{A_2}^{(true)} = \mathcal{X}_{A_2}^{(dev)} - \mathcal{X}_{A_1}^{(dev)}$$

For the test performance difference, this leads to:

$$\begin{aligned}&\Psi_{A_1}^{(test)} - \Psi_{A_2}^{(test)}\\ &= (\Psi_{A_1}^{(true)} - \Psi_{A_2}^{(true)}) + (\mathcal{X}_{A_1}^{(test)} - \mathcal{X}_{A_2}^{(test)})\\ &= (\mathcal{X}_{A_2}^{(dev)} - \mathcal{X}_{A_1}^{(dev)}) + (\mathcal{X}_{A_1}^{(test)} - \mathcal{X}_{A_2}^{(test)})\end{aligned}$$

The difference in test performance between A_1 and A_2 does not only depend on $\mathcal{X}^{(test)}$, but also on the random variable of the development set $\mathcal{X}^{(dev)}$. Hence, the variance introduced by the finite approximation of the development set is important to understand the variance of test scores.

6.2 Emperical Estimation

In this section we study how large the test score can vary for the studied tasks from section 4. We assume $\Psi_{A_1}^{(dev)} = \Psi_{A_2}^{(dev)}$. We are interested in how much the test score for these two models can vary, i.e. how large the difference $|\Psi_{A_1}^{(test)} - \Psi_{A_2}^{(test)}|$ can reasonably become.

We do this by computing a linear regression $f(\Psi^{(dev)}) \approx \Psi^{(test)}$ between the development and test score. For this linear regression, we compute the prediction interval ζ (Faraway, 2002). The test score should be within the range $f(\Psi^{(dev)}) \pm \zeta(\Psi^{(dev)})$ with a confidence of α.

The prediction interval is given by:

$$\zeta(\Psi^{(dev)}) = t_{n-2}^* s_y \sqrt{1 + \frac{1}{n} + \frac{(\Psi^{(dev)} - \overline{\Psi^{(dev)}})^2}{(n-1)s_x^2}}$$

with n the number of samples, t_{n-2}^* the value for the two-tailed t-distribution at the desired confidence α for the value $n-2$, s_y the standard deviation of the residuals calculated as:

$$s_y = \sqrt{\frac{\sum(\Psi^{(test)} - \hat{\Psi}^{(test)})^2}{n-2}}$$

$\overline{\Psi^{(dev)}}$ the mean value $\Psi_i^{(dev)}$ and s_x the unbiased estimation of standard deviation:

$$s_x^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(\Psi_i^{(dev)} - \overline{\Psi^{(dev)}}\right)^2.$$

An extreme difference in test score would be $\Psi_{A_1}^{(test)} \leq f(\Psi^{(dev)}) - \zeta(\Psi^{(dev)})$ for the one model and $\Psi_{A_2}^{(test)} \geq f(\Psi^{(dev)}) + \zeta(\Psi^{(dev)})$ for the other model. The difference would then be $|\Psi_{A_1}^{(test)} - \Psi_{A_2}^{(test)}| \geq 2\zeta(\Psi^{(dev)})$.

The probability of $|\Psi_{A_1}^{(test)} - \Psi_{A_2}^{(test)}| \geq 2\zeta(\Psi^{(dev)})$ is $(1-\alpha)^2$. We set $(1-\alpha)^2 = 0.05$. In this case, $|\Psi_{A_1}^{(test)} - \Psi_{A_2}^{(test)}| \leq 2\zeta(\Psi^{(dev)})$ in 95% of the cases.

The value of $2\zeta(\Psi^{(dev)})$ is approximately constant in terms of the development score $\Psi^{(dev)}$. Hence, we computed the mean $2\overline{\zeta(\Psi^{(dev)})}$ and depict the value in Table 3.

Task	Predict. Interval
ACE 2005 - Entities	1.03
ACE 2005 - Events	3.68
CoNLL 2000 - Chunking	0.25
CoNLL 2003 - NER-En	0.69
CoNLL 2003 - NER-De	1.24
GermEval 2014 - NER-De	0.88
TempEval 3 - Events	1.30

Table 3: Size of the 95% interval for the test scores of two models with the same development score.

The value 3.68 for the *ACE 2005 - Events* tasks indicates that, given two models with the same performance on the development set, the test performance can vary up to 3.68 percentage points F_1-score (95% interval). The values $2\overline{\zeta(\Psi^{(dev)})}$ are comparably similar to the value of $\Delta_{95}^{(test)}$ in Table 2.

7 Evaluation Methodologies based on Score Distributions

In this section, we formally define two idealized definitions for *approach A superior to approach B*.

We define the performance for a model as:

$$\Psi_{A_{(\texttt{Train},\texttt{Dev},\texttt{Rnd})}}^{(Test)} = S(A_{(\texttt{Train},\texttt{Dev},\texttt{Rnd})}(\texttt{Test}_x), \texttt{Test}_y). \quad (1)$$

A is the learning approach that trains a model given a training set `Train`, a development set `Dev` and a sequence of random numbers `Rnd`. The resulting model $A_{(\texttt{Train},\texttt{Dev},\texttt{Rnd})}$ is applied to the test dataset $\texttt{Test}_x$ and a performance score S is computed between the predictions and the gold labels $\texttt{Test}_y$.

Evaluation 3. Given a certain task and a potentially infinite data population $\mathcal{D}$. We call *approach A superior to approach B* for this task with training set of size $k \leq |\texttt{Train}| \leq l$ if and only if the expected test score for approach A is larger than the expected test score for approach B:

$$E\left[\Psi_{A_{(\texttt{Train},\texttt{Dev},\texttt{Rnd})}}^{(Test)}\right] > E\left[\Psi_{B_{(\texttt{Train},\texttt{Dev},\texttt{Rnd})}}^{(Test)}\right]$$

with `Train`, `Dev`, and `Test` sampled from $\mathcal{D}$.

We can approximate the expected test score for an approach by training multiple models and comparing the sample mean values $\overline{\Psi_{A_{1\ldots n}}^{(Test)}}$ and $\overline{\Psi_{B_{1\ldots m}}^{(Test)}}$. We conclude that one approach is superior if the difference between the means is significant.

A common significance test used in literature is the Welch's t-test. This is a simple significance test which only requires the information on the sample mean, sample variance and sample size. However, the test assumes that the two distributions are approximately normally distributed.

Evaluation 3 computes the expected test score, however, *superior* can also be interpreted as a higher probability to produce a better working model.

Evaluation 4. Given a certain task and a potentially infinite data population $\mathcal{D}$. We call *approach A superior to approach B* for this task with training set of size $k \leq |\texttt{Train}| \leq l$ if and only if the probability for approach A is higher to produce a better working model than it is for approach B. We call approach A superior to approach B if and only if:

$$P\left(\Psi_{A_{(\texttt{Train},\texttt{Dev},\texttt{Rnd})}}^{(Test)} \geq \Psi_{B_{(\texttt{Train},\texttt{Dev},\texttt{Rnd})}}^{(Test)}\right) > 0.5$$

We can estimate if the probability is significantly different from 0.5 by sampling a sufficiently large number of models from approach A and approach B and then applying either a Mann-Whitney U test for independent pairs or a Wilcoxon signed-rank test

for matched (dependent) pairs for the achieved test scores.

In contrast to the Welch's t-test, those two tests do not assume a normal distribution. To perform the Wilcoxon signed-rank test, at least 6 models for a two-tailed test are needed to be able to get a confidence level $p < 0.05$ (Sani and Todman, 2005). For a confidence level of $p < 0.01$, at least 8 models are needed.

There is a fine distinction between Evaluation 3 and Evaluation 4. Evaluation 3 compares the mean values for two approaches, while Evaluation 4 compares the medians of the distributions.[6] For skewed distributions, the median is different from the mean, which might change the drawn conclusion from Evaluation 3 and Evaluation 4. Approach A might have a better mean score than approach B, but a lower median than approach B or vice versa.

Note, `Train`, `Dev`, and `Test` in Evaluation 3 and 4 are random variables sampled from the (infinite) data population $\mathcal{D}$. This is an idealized formulation for comparing machine learning approaches as it assumes that new, independent datasets from $\mathcal{D}$ can be sampled. However, for most tasks, it is not easily possible to sample new datasets. Instead, only a finite dataset is labeled that must be used for `Train`, `Dev`, and `Test`. This creates the risk that an approach might be superior for a specific dataset, however, for other train, development, or test sets, this might not be the case. In contrast, addressing the variation introduced by `Rnd` is straightforward by training the approach with multiple random sequences.

Evaluation 3 and Evaluation 4 both mention that training sets are of size $k \leq |\texttt{Train}| \leq l$. Learning approaches can react differently to increasing or decreasing training set sizes, e.g., approach A might be better for larger training sets while approach B might be better for smaller training sets. When comparing approaches, it would be of interest to know the lower bound k and the upper bound l for approaches A and B. However, most evaluations check for practical reasons only one training set size, i.e., $k = l$.

[6] Note, for certain distributions, the median m with $P(X \leq m) \leq 0.5$ and $P(X \geq m) \leq 0.5$ might not be uniquely defined. This does not affect Evaluation 4.

8 Experiment (Score Distributions)

In this section, we study if Evaluation 3 and Evaluation 4 can reliably detect that there is no difference between approach A and $\tilde{A}$ from section 4.

We compare 25 models from approach A ($A_1^{(j)}, ..., A_{25}^{(j)}$) with 25 models from approach $\tilde{A}$ ($\tilde{A}_1^{(j)}, ..., \tilde{A}_{25}^{(j)}$) each trained with a different random sequence `Rnd`. For Evaluation 3, we use Welch's t-test, for Evaluation 4, Wilcoxon signed-rank test. As threshold, we used $p < 0.05$.

Task	Eval. 3	Eval. 4
ACE - Entities	4.68%	4.86%
ACE - Events	4.72%	4.67%
CoNLL - Chunking	4.60%	4.86%
CoNLL - NER-En	5.18%	5.01%
CoNLL - NER-De	4.83%	4.78%
GermEval - NER-De	4.91%	4.74%
TempEval - Events	4.72%	5.03%

Table 4: Percentage of significant difference between A and $\tilde{A}$ for $p < 0.05$.

Table 4 summarizes the outcome of this experiment. The ratios are all at about 5%, which is the number of false positives we would expect from a threshold $p < 0.05$. In contrast to Evaluation 1 and 2, Evaluation 3 and 4 were able to identify that the approaches are identical in most cases.

Next, we study how stable the mean $\overline{\Psi^{(test)}_{A^{(j)}_{1,...,n}}}$ is for various values of n. The larger the variance, the more difficult will it be to spot a difference between two learning approaches. To express the variance in an intuitive value, we compute the 95th percentile $\Delta^{(test)}_{95}$ for the difference between the mean scores:

$$\Delta^{(test),(n,j)} = \left| \overline{\Psi^{(test)}_{A^{(j)}_{1,...,n}}} - \overline{\Psi^{(test)}_{\tilde{A}^{(j)}_{1,...,n}}} \right|$$

The value $\Delta^{(test)}_{95}$ gives an impression which improvement in mean test score is needed for a significant difference. Note, this value depends on the variance of the produced models.

The values are depicted in Table 5. For increasing n the value $\Delta^{(test)}_{95}$ decreases, i.e. the mean score becomes more stable. However, for the CoNLL 2003

NER-En task we still observe a difference of 0.26 percentage points F_1-score between the mean scores for $n = 10$. For the ACE 2005 Events dataset, the value is even at 1.39 percentage points F_1-score.

	$\Delta_{95}^{(test)}$ **for** n **scores**				
Task	**1**	**3**	**5**	**10**	**20**
ACE-Ent.	1.21	0.72	0.51	0.38	0.26
ACE-Ev.	4.32	2.41	1.93	1.39	0.97
Chk.	0.30	0.16	0.14	0.09	0.06
NER-En	0.83	0.45	0.35	0.26	0.18
NER-De	1.61	0.94	0.72	0.51	0.37
GE 14	1.12	0.64	0.48	0.34	0.25
TE 3	1.48	0.81	0.63	0.48	0.32

Table 5: 95% percentile of $\Delta^{(test)}$ after averaging.

9 Discussion & Conclusion

Non-deterministic approaches like neural networks can produce models with varying performances and comparing performances based on single models does not allow drawing conclusions about the underlying learning approaches.

An interesting observation is that the variance of the test scores depends on the development set. With an improper development set, the achieved test scores for the same approach can vary arbitrarily large. Without a good development set, we face the challenge of not knowing which configuration in weight space to choose.

We conclude that the meaningfulness of a test score is limited by the quality of the development set. This is an important observation, as often little attention is paid to the selection of the development set. To have as much training data as possible, we often prefer small development sets, sometimes substantially smaller than the test set.

Future work is needed to judge the importance of the development set and how to select it appropriately. As of now, we recommend using a development set that is of comparable size to the test set.

For the organization of shared tasks, we recommend that participants do not submit only a single model, but multiple models trained with different random seed values. Those submissions should not be treated individually. Instead the mean and the standard deviation of test scores should be reported.

Previous work showed that there can be large differences between local minima of neural networks and that some minima generalize badly to unseen data. Those minima also generalize badly on the development set and do not play a role in the final evaluation. This form of evaluation, where only the model that performed best on the development set is evaluated on unseen test data, can be found in many publications and many shared tasks evaluate individual models submitted by the participants.

We showed that this evaluation setup is not suitable to draw conclusions about machine learning approaches. A statistically significant difference of test scores does not have to be the result of a superior learning approach. There is a high risk that this is due to chance. Further, we showed that the development set has a major impact on the test score variance.

Our observations are not limited to non-deterministic machine learning approaches. If we treat hyperparameters as part of an approach, it also affects deterministic approaches like support vector machines. For an SVM we might achieve with two slightly different configurations identical development scores, however, both models might show a large difference in terms of test score. It is up to chance which model would be select for the final evaluation.

We provide two formalizations for comparing learning approaches. The first compares expected scores, however, it requires that scores are approximately normal distributed for significance testing. The second defines superiority of a learning approach in terms of the probability to produce a better working model. This definition can be tested without the assumption of normal distributed scores. For the evaluated approach and tasks, we showed that the type I error rate matches the p-value of the significance tests.

For shared tasks, we propose that participants submit multiple models, at least 6 for a p-value of 0.05, trained with different sequences of random numbers. Those submissions should not be treated individually. Instead we recommend the comparison of score distributions.

References

Taylor Berg-Kirkpatrick, David Burkett, and Dan Klein. 2012. An Empirical Investigation of Statistical Significance in NLP. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 995–1005, Stroudsburg, PA, USA. Association for Computational Linguistics.

Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Daniel Cer, Mona Diab, Eneko Agirre, Iigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada.

Timothy Dozat. 2015. Incorporating Nesterov Momentum into Adam.

Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. 2010. Why Does Unsupervised Pre-training Help Deep Learning? *Journal of Machine Learning Research*, 11:625–660, March.

Julian J. Faraway. 2002. *Practical Regression and ANOVA using R*. University of Bath.

Yarin Gal and Zoubin Ghahramani. 2016. A Theoretically Grounded Application of Dropout in Recurrent Neural Networks. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 1019–1027.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Flat Minima. *Neural Computation*, 9(1):1–42, January.

Nathalie Japkowicz and Mohak Shah. 2011. *Evaluating Learning Algorithms: A Classification Perspective*. Cambridge University Press, New York, NY, USA.

Alexandros Komninos and Suresh Manandhar. 2016. Dependency based embeddings for sentence classification tasks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1490–1500, San Diego, California, June. Association for Computational Linguistics.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *CoRR*, abs/1603.01360.

Yann LeCun, Léon Bottou, Genevieve B. Orr, and Klaus-Robert Müller. 1998. Efficient BackProp. In *Neural Networks: Tricks of the Trade, This Book is an Outgrowth of a 1996 NIPS Workshop*, pages 9–50, London, UK, UK. Springer-Verlag.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint Event Extraction via Structured Prediction with Global Features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 73–82, Sofia, Bulgaria, August. Association for Computational Linguistics.

Xuezhe Ma and Eduard H. Hovy. 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. *CoRR*, abs/1603.01354.

Nils Reimers and Iryna Gurevych. 2017a. Optimal Hyperparameters for Deep LSTM-Networks for Sequence Labeling Tasks. *arXiv preprint arXiv:1707.06799*.

Nils Reimers and Iryna Gurevych. 2017b. Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Copenhagen, Denmark, September.

Nils Reimers, Judith Eckle-Kohler, Carsten Schnober, Jungi Kim, and Iryna Gurevych. 2014. Germeval-2014: Nested named entity recognition with neural networks. In Gertrud Faaß and Josef Ruppenhofer, editors, *Workshop Proceedings of the 12th Edition of the KONVENS Conference*, pages 117–120. Universitätsverlag Hildesheim, October.

Stefan Riezler and John T. Maxwell. 2005. On Some Pitfalls in Automatic Evaluation and Significance Testing for MT. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 57–64, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Fabio Sani and John Todman. 2005. *Experimental Design and Statistics for Psychology: A First Course*. Wiley-Blackwell.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going Deeper with Convolutions. In *Computer Vision and Pattern Recognition (CVPR)*.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajic, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinkova, Jan Hajic jr., Jaroslava Hlavacova, Václava Kettnerová, Zdenka Uresova, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria dePaiva, Kira Droganova, Héctor Martínez Alonso, Çağr Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali

Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonca, Tatiana Lando, Rattima Nitisaroj, and Josie Li. 2017. Conll 2017 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, Vancouver, Canada, August. Association for Computational Linguistics.