Deep Dominance - How to Properly Compare Deep Neural Models

Rotem Dror **Segev Shlomov** **Roi Reichart**
Faculty of Industrial Engineering and Management, Technion, IIT
rtmdrr|segevs|roiri@technion.ac.il

Abstract

Comparing between Deep Neural Network (DNN) models based on their performance on unseen data is crucial for the progress of the NLP field. However, these models have a large number of hyper-parameters and, being non-convex, their convergence point depends on the random values chosen at initialization and during training. Proper DNN comparison hence requires a comparison between their empirical score distributions on unseen data, rather than between single evaluation scores as is standard for more simple, convex models. In this paper, we propose to adapt to this problem a recently proposed test for the *Almost Stochastic Dominance* relation between two distributions. We define the criteria for a high quality comparison method between DNNs, and show, both theoretically and through analysis of extensive experimental results with leading DNN models for sequence tagging tasks, that the proposed test meets all criteria while previously proposed methods fail to do so. We hope the test we propose here will set a new working practice in the NLP community.[1]

1 Introduction

A large portion of the research activity in Natural Language Processing (NLP) is devoted to the development of new algorithms for existing or new tasks. To evaluate the quality of a new method, its performance on unseen datasets is compared to the performance of existing methods. The progress of the field hence crucially depends on our ability to draw conclusions from such comparisons.

In the past, most supervised NLP models have been linear (or log-linear), convex and relatively simple (e.g. (Toutanova et al., 2003; Finkel et al., 2008; Ritter et al., 2011)). Hence, their training was deterministic and the number of configurations a model could have was rather small – decisions about model design were usually limited to feature selection and the selection of one of a few loss functions. Consequently, when one model performed better than another on unseen data it was safe to argue that the winning model was generally better, especially when the results were statistically significant (Dror et al., 2018), and when the effect of multiple hypotheses testing was taken into account in cases of evaluation with multiple datasets (Dror et al., 2017).

With the recent emergence of Deep Neural Networks (DNNs), data-driven performance comparison has become much more complicated. While models such as LSTM (Hochreiter and Schmidhuber, 1997), Bi-LSTM (Schuster and Paliwal, 1997) and the transformer (Vaswani et al., 2017) improved the state-of-the-art in many NLP tasks (e.g. (Dozat and Manning, 2017; Hershcovich et al., 2017; Yadav and Bethard, 2018)), it is much more difficult to compare the performance of algorithms that are based on these models. This is because the loss functions of these models are non-convex (Dauphin et al., 2014), making the solution to which they converge (a local minimum or a saddle point) sensitive to random weight initialization and the order of training examples. Moreover, as these complex models are not fully understood, their training is often enhanced by heuristics such as random dropouts (Srivastava et al., 2014) that introduces another level of non-determinism to the training process. Finally, the increased model complexity results in a much larger number of configurations, governed by a large space of hyper-parameters for model properties such as the number of layers and the number of neurons in each layer.

With so many degrees of freedom governed by random and arbitrary values, when comparing two

[1] Our code is available at: https://github.com/rtmdrr/deepComparison

DNNs it is not possible to consider a single test-set evaluation score for each model. If we do that, we might compare just the best models that someone happened to train rather than the methods themselves. Instead, it is necessary to compare between the score distributions generated by different runs of each of the models. Unfortunately, this comparison task, which is fundamental to the progress of the field, has not received a systematic treatment thus far. Our goal is to close this gap and propose a simple and effective comparison tool between two DNNs based on their test set score distributions. Particularly, we make four contributions:

Defining a DNN comparison framework: We define three criteria that a DNN comparison tool should meet: **(a)** Since we observe only a sample from the population score distribution of each model, the decision should be *significant* under well justified statistical assumptions. This assures that future runs of the superior model are likely to get higher scores than future runs of the inferior model; **(b)** The decision mechanism should be *powerful*, being able to make decisions in most possible decision tasks; and, finally, **(c)** Since both models depend on random decisions, it is likely that none of them is promised to be superior over the other in all cases (e.g. with all possible random seeds). A powerful comparison tool should hence augment its decision with a *confidence score*, reflecting the probability that the superior model will indeed produce a better output.

Analysis of existing solutions (§ 3, 5): The comparison problem we address has been highlighted by Reimers and Gurevych (2017b, 2018), who established its importance in an extensive experimentation with neural sequence models (Reimers and Gurevych, 2017a), and proposed two main solutions (§3). One solution, which we refer to as the *collection of statistics (COS)* solution, is based on the analysis of statistics of the empirical score distribution of the two algorithms – such as their mean, median and standard deviation (std), as well as their minimum and maximum values. Unfortunately, this solution does not respect criterion (a) as it does not deal with significance, and as we demonstrate in §5 its power (criterion (b)) is also limited. Their second solution is based on significance testing for *Stochastic Order (SO)* (Lehmann, 1955), a strict criterion that is hardly met in reality. While this solution respects criterion (a), it is not designed to deal with criterion (c), since it does not provide information beyond its decision if one of the distributions is stochastically dominant than the other or not, and as we show in §5 its power (criterion (b)) is very limited.

A new comparison tool (§ 4): We propose a solution that meets our three criteria. Particularly, we adapt to our problem the recently presented concept of *Almost Stochastic Order (ASO)* between two distributions (Álvarez-Esteban et al., 2017),[2] and the statistical significance test for this property, which makes very modest assumptions about the participating distributions (criterion (a)). Further, in line with criterion (c), the test returns a variable $\epsilon \in [0, 1]$, that quantifies the degree to which one algorithm is stochastically larger than the other, with $\epsilon = 0$ reflecting stochastic order. We further show that the test is designed to be very powerful (criterion (b)), which is possible because the decision on the superior algorithm is complemented by the confidence score.

Extensive experimental analysis (§ 5): We revisit the extensive experimental setup of Reimers and Gurevych (2017a,b), who performed 510 comparisons between strong DNN-based sequence tagging models. In each of their experiments they compared two models – either different models or two variants of the same model differing in some of their hyper-parameters – and reported the score distributions of each model across various random seeds and hyper-parameter configurations. Our analysis reveals that while our test can declare one of the algorithms superior in 100% of the cases, the COS approach can do that in 49.01% of the cases, and the SO approach in a mere 0.98%. In addition to being powerful, the decisions and the confidence scores of our proposed test are also well aligned with the tests proposed in previous literature: when the previous methods are challenged, our method still makes a decision but it also indicates the smaller gap between the algorithms. We hope that this work will establish a standard for the comparison between DNNs.

2 Performance Variance in DNNs

In this section we discuss the source of non-determinism in DNNs, focusing on hyper-parameter configurations and random choices.

Hyper-parameter Configurations DNNs are complex models governed by a variety of hyper-

[2] We use the terms Almost Stochastic Order and Almost Stochastic Dominance interchangeably in this paper.

parameters. A formal definition of a hyper-parameter, differentiating it from a standard parameter, is usually a parameter whose value is set before the learning process begins. We can roughly say that hyper-parameters determine the structure of the model and particular algorithmic decisions related, e.g., to its optimization. Some popular structure-related hyper-parameters in the DNN literature are the number of layers, layer sizes, activation functions, loss functions, window sizes, stride values, and parameter initialization methods. Some optimization (training) related hyper-parameters are the optimization algorithms, learning rates, number of epochs, momentum, mini-batch sizes, whether or not to use optimization heuristics such as gradient clipping and gradient normalization, and sampling and ordering methods of the training data.

To decide on the hyper-parameter values, it is standard to explore several configurations and observe which performs best on an unseen, held-out dataset, commonly referred to as the development set. For some hyper-parameters (e.g. the learning rate and the optimization algorithm) the range of feasible values reflects the intuitions of the model author and the tuned value provides some insight about the model and the data. However, for many other hyper-parameters (e.g. the number of neurons in each layer of the model and the number of epochs of the optimization algorithm) the range of values and the selected values are quite arbitrary. Hence, although hyper-parameters can be tuned on development data, the distribution of model's evaluation scores across these configurations is of interest, especially when considering the hyper-parameters with the more arbitrary values.

Random Choices There are also hyper-parameters that do not follow the above tuning logic. These include some of the hyper-parameters that govern the random ordering of the training examples, the dropout process and the initialization of the model parameters. The values of these hyper-parameters are often set randomly.

In other cases, randomization is introduced to the model without an explicit hyper-parameter. For example, a popular initialization method for DNN weights is the Xavier method (Glorot and Bengio, 2010). In this method, the initial weights are sampled from a Gaussian distribution with a mean of 0 and an std of $\sqrt{2/n_i}$, where n_i is the number of input units of the i-th layer.

As discussed in §1, being non-convex, the convergent point of DNNs is deeply affected by these random effects. Unfortunately, exploring all possible random seeds is impossible both because they form an uncountable set and because their values are uninterpretable and it is hence even hard to decide on the relevant search space for their values. This dictates the need for reporting model results with multiple random choices.

3 Comparing DNNs: Problem Formulation and Background

Problem Definition Given two algorithms, each associated with a set of test-set evaluation scores, our goal is to determine which algorithm, if any, is superior. In this research, the score distributions are generated when running two different DNNs with various hyper-parameter configurations and random seeds. For both DNNs, the performance is measured using the same evaluation measure over the same dataset,[3] but, to be as general as possible, the number of scores may vary between the DNNs.

As noted in §1, several methods were proposed for the comparison between the score distributions of two DNNs. We now discuss these methods.

3.1 Collection of Statistics (COS)

This approach is based on the analysis of statistics of the empirical score distributions. For example, Reimers and Gurevych (2018) averaged the test-set scores and applied the Welch's t-test (Welch, 1947) for comparing between the means. Notice that the Welch's t-test is based on the assumption that the test-set scores are drawn from normal distributions – an assumption that has not been validated for DNN score distributions. Hence, this method does not meet criterion (a) from §1, that requires the comparison method to check for statistical significance under realistic assumptions.

Moreover, comparing only the mean of two distributions is not always sufficient for making predictions about future comparisons between the algorithms. Other statistics such as the std, median and the minimum and maximum values are often also relevant. For example, it might be that the expected value of algorithm A is indeed larger than that of algorithm B, but A's std is also much larger, making prediction very challenging. In §5 we show that if both larger mean and smaller std

[3] Without loss of generality we will assume that higher values of the measure are better.

is required for a decision, the COS approach is decisive (i.e. it can declare that one algorithm is better than the other) in only 49.01% of the 510 setups considered in Reimers and Gurevych (2017b). This violates our criterion (b) which requires the comparison test to be powerful.

3.2 Stochastic Order (SO)

Another approach, proposed by Reimers and Gurevych (2018), tests whether a score drawn from the distribution of algorithm A (denoted as X_A) is likely, with a probability higher than 0.5, to be larger than a score drawn from the distribution of algorithm B (X_B). Put it formally, algorithm A is declared superior to algorithm B if:

$$P(X_A \geq X_B) > 0.5. \tag{1}$$

To test if this requirement holds based on the empirical score distributions of the two algorithms, the authors applied the Mann-Whitney U test for independent pairs (Mann and Whitney, 1947) – which tests whether there exists a stochastic order (SO) between two random variables. This test is non-parametric, making no assumptions about the participating distributions except for being continuous. In the appendix we show that if there is an SO between two distributions, the condition in Equation 1 also holds.

We next describe the concept of SO in more details. But first, in order to keep our paper self-contained, we define the cumulative distribution function (CDF) and the empirical CDF of a probability distribution.

The CDF For a random variable X, the CDF is defined as follows:

$$F(t) = P(X \leq t).$$

For a sample $\{x_1, .., x_n\}$, the empirical CDF is defined as follows:

$$F_n(t) = \frac{1}{n}\sum_{i=1}^{n} 1_{x_i \leq t} = \frac{\text{number of } x_i\text{s} \leq t}{n},$$

where $1_{x_i \leq t}$ is an indicator function that takes the value of 1 if $x_i \leq t$, and 0 otherwise. These definitions are required for the definitions of SO we make next.

Stochastic Order (SO) Lehmann (1955) defines a random variable X to be *stochastically larger* than a random variable Y (denoted by $X \succeq Y$) if $F(a) \leq G(a)$ for all a (with a strict inequality for some values of a), where F and G are the CDFs of X and Y, respectively. That is, if we observe a random value sampled from the first distribution, it is likely to be larger than a random value sampled from the second distribution.

If it can be concluded from the empirical score distributions of two DNNs that SO exists between their respective population distributions, this means that one algorithm is more likely to produce higher quality solutions than the other, and this algorithm can be declared superior. As discussed above, Reimers and Gurevych (2018) applied the Mann-Whitney U-test to test for this relationship. The U-test has high statistical power when the tested distributions are moderate-tailed, e.g., the normal distribution or the logistic distribution. When the distribution is heavy tailed, e.g., the Cauchy distribution, there are several alternative statistical tests that have higher statistical power, for example likelihood based tests (Lee and Wolfe, 1976; El Barmi and McKeague, 2013).

The main limitation of this approach is that SO can rarely be proved to hold based on two empirical distributions. Indeed, in §5 we show that an SO holds between the two compared algorithms only in 0.98% of the comparisons performed by Reimers and Gurevych (2017a). Hence, while this approach meets our criterion (a) (testing for significance under realistic assumptions), it does not meet criterion (b) (being a powerful test) and criterion (c) (providing a confidence score).

We will next describe another approach that does meet our three criteria.

4 Our Approach: Almost Stochastic Dominance

Our starting point is that the requirement of SO is unrealistic because it means that the inequality $F(a) \leq G(a)$ should hold for every value of a. It is likely that this criterion should fail to determine dominance between two distributions even when a ”reasonable” decision-maker would clearly prefer one DNN over the other. We hence propose to employ a relaxed version of this criterion. We next discuss different definitions of such relaxation.

A Potential Relaxation For $\epsilon > 0$ and random variables X and Y with CDFs F and G, respectively, we can define the following notion of ϵ-stochastic dominance:

$$X \succeq_\epsilon Y \text{ if } F(a) \leq G(a) + \epsilon \text{ for all } a.$$

That is, we allow the distributions to violate the stochastic order, and hence one CDF does not have to be strictly below the other for all a.

The practical shortcomings of this definition are apparent in cases where $F(a)$ is greater than $G(a)$ for all a, with a gap bounded by, for example, $\epsilon/2$. In such cases we would not want to determine that $X \sim F$ is ϵ stochastically dominant over $Y \sim G$ because its CDF is strictly above the CDF of Y, and hence Y is stochastically larger than X. However, according to this relaxation, $X \sim F$ is indeed ϵ stochastically larger than $Y \sim G$.

Almost Stochastic Dominance To overcome the limitations of the above straight forward approach, and define a relaxation of stochastic order, we turn to a definition that is based on the proportion of points in the domain of the participating distributions for which SO holds. That is, the test we will introduce below is based on the following two violation sets:

$$V_X = \{a : F(a) > G(a)\}.$$

$$V_Y = \{a : F(a) < G(a)\}.$$

Intuitively, the variable with the smaller violation set should be declared superior and the ratio between these sets should define the gap between the distributions.

To implement this idea, del Barrio et al. (2018) defined the concept of *almost stochastic dominance*. Here we describe their work, that aims to compare two distributions, and discuss its applicability to our problem of comparing two DNN models based on the three criteria defined in §1. We start with a definition: for a CDF F, the *quantile function* associated with F is defined as:

$$F^{-1}(t) = \inf\{x : t \leq F(x)\}, t \in (0,1). \quad (2)$$

It is possible to define stochastic order using the quantile function in the following manner:

$$X \succeq Y \iff F^{-1}(t) \geq G^{-1}(t), \forall t \in (0,1). \quad (3)$$

The advantage of this definition is that the domain of the quantile function is bounded between 0 and 1. This is in contrast to the CDF whose domain is unbounded.

From this definition, it is clear that a violation of the stochastic order between X and Y occurs when $F^{-1}(t) < G^{-1}(t)$. Hence, it is easy to redefine V_X and V_Y based on the quantile functions:

$$A_X = \{t \in (0,1) : F^{-1}(t) < G^{-1}(t)\}.$$

$$A_Y = \{t \in (0,1) : F^{-1}(t) > G^{-1}(t)\}.$$

del Barrio et al. (2018) employed these definitions in order to define the distance of each random variable from stochastic dominance over the other:

$$\varepsilon_{\mathcal{W}_2}(F,G) := \frac{\int_{A_X} (F^{-1}(t) - G^{-1}(t))^2 dt}{\big(W_2(F,G)\big)^2}. \quad (4)$$

Where $W_2(F,G)$, also known as the univariate L_2-Wasserstein distance between distributions, is defined as:

$$W_2(F,G) = \sqrt{\int_0^1 (F^{-1}(t) - G^{-1}(t))^2 dt}. \quad (5)$$

This ratio explicitly measures the distance of X (with CDF F) from stochastic dominance over Y (with CDF G) since it reflects the probability mass for which Y dominates X. The corresponding definition for the distance of Y from being stochastically dominant over X can be received from the above equations by replacing the roles of F and G and integrating over A_Y instead of A_X.

This index satisfies $0 \leq \varepsilon_{\mathcal{W}_2}(F,G) \leq 1$ where 0 corresponds to perfect stochastic dominance of X over Y and 1 corresponds to perfect stochastic dominance of Y over X. It also holds that $\varepsilon_{\mathcal{W}_2}(F,G) = 1 - \varepsilon_{\mathcal{W}_2}(G,F)$, and smaller values of the smaller index (which is by definition bounded between 0 and 0.5) indicate a smaller distance from stochastic dominance.

Statistical Significance Testing for ASO Using this index it is possible to formulate the following hypothesis testing problem to test for almost stochastic dominance:

$$H_0 : \varepsilon_{\mathcal{W}_2}(F,G) \geq \epsilon$$

$$H_1 : \varepsilon_{\mathcal{W}_2}(F,G) < \epsilon$$

which tests, for a predefined $\epsilon > 0$, if the violation index is smaller than ϵ. Rejecting the null hypothesis means that the first score distribution F is almost stochastically larger than G with ϵ distance from stochastic order.

del Barrio et al. (2018) proved that without further assumptions, H_0 will be rejected with a significance level of α if:

$$\sqrt{\frac{nm}{n+m}}\big(\varepsilon_{W_2}(F_n, G_m) - \epsilon\big) < \hat{\sigma}_{n,m}\Phi^{-1}(\alpha),$$

where F_n, G_m are the empirical CDFs with n and m samples, respectively, ϵ is the violation level, Φ^{-1} is the inverse CDF of a normal distribution and $\hat{\sigma}_{n,m}$ is the estimated variance of the value

$$\sqrt{\frac{nm}{n+m}}\big(\varepsilon_{\mathcal{W}_2}(F_n^*, G_m^*) - \varepsilon_{\mathcal{W}_2}(F_n, G_m)\big),$$

where $\varepsilon_{\mathcal{W}_2}(F_n^*, G_m^*)$ is computed using samples X_n^*, Y_m^* from the empirical distributions F_n and G_m.[4]

In addition, the minimal ϵ for which we can claim, with a confidence level of $1-\alpha$, that F is almost stochastically dominant over G is $\epsilon^{min}(F_n, G_m, \alpha) =$ $\varepsilon_{W_2}(F_n, G_m) - \sqrt{\frac{n+m}{nm}}\hat{\sigma}_{n,m}\Phi^{-1}(\alpha)$.

If $\epsilon^{min}(F_n, G_m, \alpha) < 0.5$, we can claim that algorithm A is better than B, and the lower $\epsilon^{min}(F_n, G_m, \alpha)$ is the greater is the gap between the algorithms. When $\epsilon^{min}(F_n, G_m, \alpha) = 0$, algorithm A is stochastically dominant over B. However, if $\epsilon^{min}(F_n, G_m, \alpha) \geq 0.5$, then F is not almost stochastically larger than G (with confidence level $1-\alpha$) and hence we should accept the null hypothesis that algorithm A is not superior to algorithm B.

del Barrio et al. (2018) proved that, assuming accurate estimation of $\hat{\sigma}_{n,m}$, it holds that:

$$\epsilon^{min}(F_n, G_m, \alpha) = 1 - \epsilon^{min}(G_m, F_n, \alpha).$$

Hence, for a given α value, one of the algorithms will be declared superior, unless $\epsilon^{min}(F_n, G_m, \alpha) = \epsilon^{min}(G_m, F_n, \alpha) = 0.5$.

Notice that the minimal ϵ and the rejection condition of the null hypothesis depend on n and m, the number of scores we have for each algorithm. Hence, for the statistical test to have high statistical power we need to make sure that n and m are big enough. While we cannot provide a method for tuning these numbers, we note that in the extensive analysis of §5 the test had enough statistical power to make decisions in all cases. The pseudo code of our implementation is provided in the appendix.

To summarize, the test for almost stochastic dominance meets the three criteria defined in §1. This is a test for statistical significance under very minimal assumptions on the distribution from which the performance scores are drawn (criterion (a)). Moreover, it quantifies the gap between the two reference distributions (criterion (c)), which allows it to make decisions even in comparisons where the gap between the superior algorithm and the inferior algorithm is not large (criterion (b)).

To demonstrate the appropriateness of this method for the comparison between two DNNs we next revisit the extensive experimental setup of Reimers and Gurevych (2017a).

5 Analysis

Tasks and Models In this section we demonstrate the potential impact of testing for almost stochastic dominance on the way empirical results of NLP models are analyzed. We use the data of Reimers and Gurevych (2017a)[5] and Reimers and Gurevych (2017b).[6] This data contains 510 comparison setups for five common NLP sequence tagging tasks: Part Of Speech (POS) tagging with the WSJ corpus (Marcus et al., 1993), syntactic chucking with the CoNLL 2000 data (Sang and Buchholz, 2000), Named Entity Recognition with the CoNLL 2003 data (Sang and De Meulder, 2003), Entity Recognition with the ACE2005 data (Walker et al., 2006), and event detection with the TempEval3 data (UzZaman et al., 2013). In each setup two leading DNNs, either different architectures or variants of the same model but with different hyper-parameter configurations, are compared across various choices of random seeds and hyper-parameter configurations. The exact details of the comparisons are beyond the scope of this paper; they are documented in the above papers.

For each experimental setup, we report the outcome of three alternative comparison methods: collection of statistics (COS), stochastic order (SO), and almost stochastic order (ASO). For COS, we report the mean, std, and median of the scores for each algorithm, as well as their minimum and maximum values. We consider one algorithm to be superior over another only if both its mean is greater and its std is smaller. For SO, we employ the U-test as proposed by Reimers and Gurevych (2018), and consider a result significant if $p \leq 0.05$. Finally, for ASO we employ the method of §4 and report the identity of the superior algorithm along with its ϵ value, using $p \leq 0.01$.

[4] The more samples, the better. In our implementation we employ the inverse transform sampling method to generate samples.

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

[6] Which was generously given to us by the authors.

Analysis Structure We divide our analysis into three cases. In *Case A* both the COS and the SO approaches indicate that one of the models is superior. In *Case B*, the previous methods reach contradicting conclusions: while COS indicates that one of the algorithms is superior, SO comes insignificant. Finally, in *Case C* both COS and SO are indecisive. In the 510 comparisons we analyze there is no setup where SO was significant but COS could not reach a decision. We start with an example setup for each case and then provide a summary of all 510 comparisons.

Results: Case A We demonstrate that if algorithm A is stochastically larger than algorithm B then all three methods agree that algorithm A is better than B. As an example setup we analyze the comparison between the NER models of Lample et al. (2016) and Ma and Hovy (2016) when running both algorithms multiple times, changing only the random seed fed into the random number generator (41 scores from (Lample et al., 2016), 87 scores from (Ma and Hovy, 2016)). The evaluation measure is F1 score. The collection of statistics for the two models is presented in Table 1.

	Lample et al.	Ma&Hovy
Mean	0.9075	0.9056
STD	0.2237	0.3211
Median	0.9080	0.9063
Min	0.9018	0.8853
Max	0.9113	0.9100

Table 1: NER results. (Case A).

The U-test states that (Lample et al., 2016) is stochastically larger than (Ma and Hovy, 2016) with a p-value of 0.00025. This result is also consistent with the prediction of the COS approach as (Lample et al., 2016) is better than (Ma and Hovy, 2016) both in terms of mean (larger) and std (smaller). Finally, the minimum ϵ value of the ASO method is 0, which also reflects an SO.

Results: Case B We demonstrate that if the measures of mean and std from the COS approach indicate that algorithm A is better than algorithm B but stochastic dominance does not hold, then it also holds that A is almost stochastically larger than B with a small $\epsilon > 0$. As an example case we consider the experiment where the performance of a BiLSTM POS tagger with one of two optimizers, Adam (Kingma and Ba, 2014) (3898 scores) or RMSProp (Hinton et al., 2012) (1822 scores), are compared across various hyper-parameter configurations and random seeds. The evaluation measure is word level accuracy. The COS for the two models is presented in Table 2.

	Adam	RMSprop
Average	0.9224	0.9190
STD	0.0604	0.0920
Median	0.9319	0.9349
Min	0.1746	0.1420
Max	0.9556	0.9573

Table 2: POS tagging results (Case B).

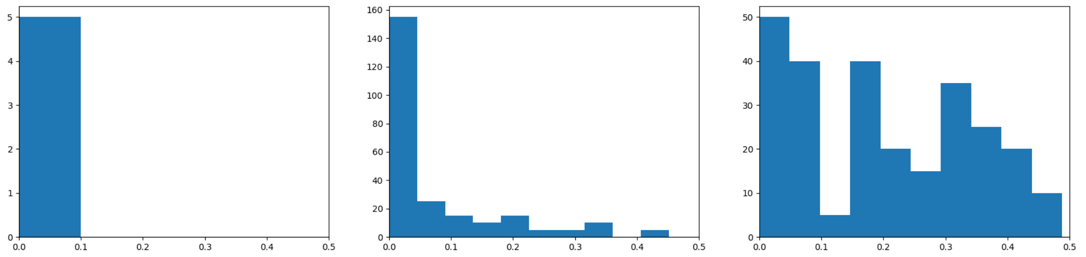
The result of the U-test came insignificant with p-value of 0.4562. The COS approach predicts that Adam is the better optimizer as both its mean is larger and its std is smaller. When comparing between Adam and RMSProrp, the ASO method returns an ϵ of 0.0159, indicating that the former is almost stochastically larger than the latter.

We note that decisions with the COS method are challenging as it potentially involves a large number of statistics (five in this analysis). Our decision here is to make the COS prediction based on the mean and std of the score distribution, even when according to other statistics the conclusion might have been different. We consider this ambiguity an inherent limitation of the COS method.

Results: Case C Finally, we address the case where stochastic dominance does not hold and no conclusions can be drawn from the statistics collection. Our observation is that even in these cases ASO is able to determine which algorithm is better with a reasonable level of confidence. We consider again a BiLSTM architecture, this time for NER, where the comparison is between two dropout policies – no dropout (225 scores) and variational dropout (2599 scores). The evaluation measure is the F1 score and the collection of statistics is presented in Table 3.

	Variational	No Dropout
Mean	0.8850	0.8772
STD	0.0392	0.0247
Median	0.8896	0.8799
Min	0.0119	0.5547
Max	0.9098	0.8995

Table 3: NER Results (Case C).



(a) Case A

(b) Case B

(c) Case C

Figure 1: An histogram of ϵ values of the ASO method for cases A, B and C.

The U-test came insignificant with a p-value of 0.5. COS is also inconclusive as the mean result of the variational dropout approach is larger, but so also its std. In this case, looking at the other statistics also gives a mixed picture as the median and max values of the variational approach are larger, but its min value is substantially smaller.

The ASO approach indicates that the no dropout approach is almost stochastically larger, with $\epsilon = 0.0279$. An in-depth consideration supports this decision as the much larger std and the much smaller minimum of the variational approach are indicators of a skewed score distribution that leaves low certainty about future performance.

Results: Summary We now turn to a summary of our analysis across the 510 comparisons of Reimers and Gurevych (2017a). Table 4 presents the percentage of comparisons that fall into each category, along with the average and std of the ϵ value of ASO for each case (all ASO results are significant with $p \leq 0.01$). Figure 1 presents the histogram of these ϵ values in each case.

	% of comparisons	Avg. ϵ	ϵ std
Case A	0.98%	0.0	0.0
Case B	48.04%	0.072	0.108
Case C	50.98%	0.202	0.143

Table 4: Results summary over the 510 comparisons of Reimers and Gurevych (2017a).

The number of comparisons that fall into case A is only 0.98%, indicating that it is rare that a decision about stochastic dominance of one algorithm can be reached when comparing DNNs. We consider this a strong indication that the Mann Whitney U test is not suitable for DNN comparison as it has very little statistical power (criterion (b)).

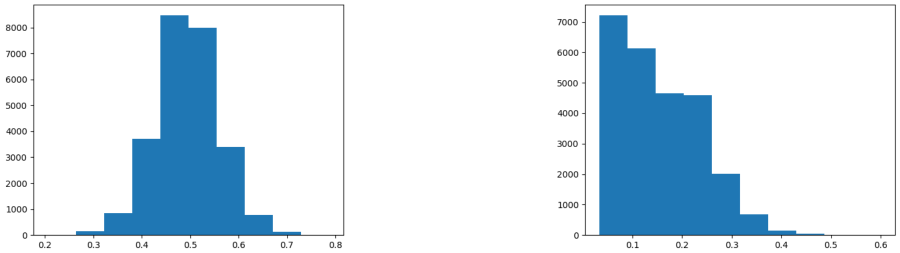
COS makes a decision in 49.01% of the comparisons (case A and B). This method is also somewhat powerful (criterion (a)), but much less so than ASO that is decisive in all 510 comparisons. The ϵ values of ASO are higher for case B than for case A (middle line of the table, middle graph of the figure). For case C the ϵ distribution is qualitatively different – ϵ receives a range of values (rightmost graph of the figure) and its average is 0.202 (bottom line of the table). We consider this to be a desired behavior as the more complex the picture drawn by COS and SO is, the less confident we expect ASO to be. Being able to make a decision in all 510 comparisons while quantifying the gap between the distributions, we believe that ASO is an appropriate tool for DNN comparison.

6 Error Rate Analysis

While our extensive analysis indicates the quality of the ASO test, it does not allow us to estimate its false positive and false negative rates. This is because in our 510 comparisons there is no oracle (or gold standard) that says if one of the algorithms is superior. Below we provide such analyses.

False Positive Rate The ASO test is defined such that the ε value required for rejecting the conclusion that algorithm A is better than B is defined by the practitioner. While $\varepsilon = 0.5$ indicates a clear rejection, most researchers would probably set a lower ε threshold. Our goal in the next analysis is to present a case where the false positive rate of ASO is very low, even when one refrains from declaring one algorithm as better than the other only when ε is very close to 0.5.

To do that, we consider a scenario where each of the 255 score distributions of the experiments in § 5 is compared to a variant of the same distribution after a Gaussian noise with a 0 expectation and a standard deviation of 0.001 is added to



(a) False Positive Rate Experiment

(b) False Negative Rate Experiment

Figure 2: Histograms of the ϵ values of the ASO test in the ablation experiments.

each of the scores. Since in all the tasks we consider the scores are in the $[0, 1]$ range, the value of 0.001 is equivalent to 0.1%. Since the average of the standard deviations of these 255 score distributions is 0.06, our noise is small but not negligible. We choose this relatively small symmetric noise so that with a high probability the original score distribution and the modified one should not be considered different. We run 100 comparisons for each of the 255 algorithms.

We compute the ε such that a value of 0 means that the non-noisy version is better than the noisy one with the strongest confidence, while the value of 1 means the exact opposite (both values are not observed in practice). A value of 0.5 indicates that no algorithm is superior – the correct prediction.

Figure 2 (a) presents a histogram of the ε values. The averaged ε is 0.502 with a standard deviation of 0.0472, and 95% of the ε values are in $[0.396, 0.631]$. This means that if we set a threshold of 0.4 on ε (i.e. lower than 0.4 or higher than 0.6), the false positive rate would be lower than 5%. In comparison, the COS approach declares the noisy version superior in 26.2% of the 255 comparisons, and the non-noisy version in 23.8%: a false positive rate of 50%.[7] The SO test makes no mistakes, as a false positive of this test is equivalent to an ε value of 0 or 1 for ASO.

Finally, we also considered a setup where for each of the 255 algorithms the performance score set was randomly split into two equal sized sets. We repeated this process 100 times for each algorithm, using ASO to compare between the sets. In all cases we observed an averaged ε of 0.5, indicating that the method avoids false positive predictions when an algorithm is compared to itself.

False Negative Rate This analysis complements the previous one by demonstrating the low false negative rate of ASO in a case where it is clear that one distribution is better than the other. For each of the 255 score distributions we generate a noisy distribution by randomly splitting the scores into a set A of $\frac{1}{4}$ of the scores and the complementary set $\hat{A}$ of the rest of the scores. For each score s we sample a noise parameter ϕ from a Gaussian with a 0 expectation and an std of 0.01, adding to s the value of $(-1) \cdot \phi^2$ if $s \in A$, and ϕ^2 if $s \in \hat{A}$. The noisy distribution is superior to the original one, with a high probability. As before we perform 100 comparisons for each of the 255 algorithms.

We compute ε such that a value of 0 would mean that the noisy version is superior. The ε values are plotted in Figure 2 (b): their average is 0.134, standard deviation is 0.07 and more than 99% of the values are lower than 0.4 (the same threshold as in the first experiment). The COS test deems the noisy distribution superior in 87.4% of the cases, while in the rest it considers none of the distributions superior. SO has a false negative rate of 100% as $\varepsilon > 0$ in all experiments.

7 Conclusions

We considered the comparison of two DNNs based on their test-set score distribution. We defined three criteria for a high quality comparison method, demonstrated that previous methods do not meet these criteria and proposed to use the recently proposed test for almost stochastic dominance that does meet these criteria. We analyzed the extensive experimental setup of Reimers and Gurevych (2017a) and demonstrated the effectiveness of our proposed test. Having released our code, we hope this will become a new evaluation standard in the NLP community.

[7] Recall that we consider one algorithm superior over the other according to COS when both the mean of its scores is larger than the mean of the other, and its std is smaller.

References

PC Álvarez-Esteban, Eustasio del Barrio, Juan Antonio Cuesta-Albertos, C Matrán, et al. 2017. Models for the assessment of treatment improvement: the ideal and the feasible. *Statistical Science*, 32(3):469–485.

Eustasio del Barrio, Juan A Cuesta-Albertos, and Carlos Matrán. 2018. An optimal transportation approach for assessing almost stochastic order. In *The Mathematics of the Uncertain*, pages 33–44. Springer.

Yann N Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. 2014. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in neural information processing systems*, pages 2933–2941.

Timothy Dozat and Christopher D Manning. 2017. Deep biaffine attention for neural dependency parsing. In *Proc. of ICLR*.

Rotem Dror, Gili Baumer, Marina Bogomolov, and Roi Reichart. 2017. Replicability analysis for natural language processing: Testing significance with multiple datasets. *Transactions of the Association for Computational Linguistics*, 5:471–486.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhikers guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1383–1392.

Hammou El Barmi and Ian W McKeague. 2013. Empirical likelihood-based tests for stochastic ordering. *Bernoulli: official journal of the Bernoulli Society for Mathematical Statistics and Probability*, 19(1):295.

Jenny Rose Finkel, Alex Kleeman, and Christopher D Manning. 2008. Efficient, feature-based, conditional random field parsing. *Proceedings of ACL-08: HLT*, pages 959–967.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256.

Daniel Hershcovich, Omri Abend, and Ari Rappoport. 2017. A transition-based directed acyclic graph parser for UCCA. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1127–1138.

Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. 2012. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, page 14.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Young Jack Lee and Douglas A Wolfe. 1976. A distribution-free test for stochastic ordering. *Journal of the American Statistical Association*, 71(355):722–727.

Erich Leo Lehmann. 1955. Ordered families of distributions. *The Annals of Mathematical Statistics*, pages 399–419.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*.

Henry B Mann and Donald R Whitney. 1947. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pages 50–60.

Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank.

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*, pages 338–348.

Nils Reimers and Iryna Gurevych. 2018. Why comparing single performance scores does not allow to draw conclusions about machine learning approaches. *arXiv preprint arXiv:1803.09578*.

Alan Ritter, Sam Clark, Oren Etzioni, et al. 2011. Named entity recognition in tweets: an experimental study. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1524–1534. Association for Computational Linguistics.

Erik F Sang and Sabine Buchholz. 2000. Introduction to the conll-2000 shared task: Chunking. *arXiv preprint cs/0009008*.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 conference of the North American chapter of the association for computational linguistics on human language technology-volume 1*, pages 173–180. Association for Computational Linguistics.

Naushad UzZaman, Hector Llorens, Leon Derczynski, James Allen, Marc Verhagen, and James Pustejovsky. 2013. Semeval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, volume 2, pages 1–9.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus. *Linguistic Data Consortium, Philadelphia*, 57.

Bernard L Welch. 1947. The generalization ofstudent's' problem when several different population variances are involved. *Biometrika*, 34(1/2):28–35.

Vikas Yadav and Steven Bethard. 2018. A survey on recent advances in named entity recognition from deep learning models. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2145–2158.

A Proof - Equivalent Definitions of Stochastic Order

As discussed in §3, our goal here is to prove that if a random variable X is stochastically larger than a random variable Y (denoted by $X \succeq Y$), then it also holds that $P(X \geq Y) > 0.5$. This lemma explains why Reimers and Gurevych (2018) employed the Mann-Whitney U test that tests for stochastic order, while their requirement for stating that one algorithm is better than the other was that $P(X \geq Y) > 0.5$ (where X is the score distribution of the superior algorithm and Y is the score distribution of the inferior algorithm).

Lemma 1. *If $X \succeq Y$ then $P(X \geq Y) > 0.5$.*

Proof. For every two continuous random variables X, Y it holds that:

$$P(X \geq Y) + P(Y > X) = 1.$$

Let us first assume that X and Y are i.i.d and continuous. If this is the case then:

$$\begin{aligned} P(X \geq Y) + P(Y > X) &= 1 \\ P(X \geq Y) + P(X > Y) &= 1 \\ 2P(X \geq Y) &= 1 \\ P(X \geq Y) &= 0.5. \end{aligned}$$

The first pass is true because X and Y are identically distributed and the second pass is true because X and Y are continuous random variables.

Assuming that the density functions of the random variables X and Y exist (which is true because they are continues variables), we can write $P(X \geq Y)$ in the following manner:

$$\begin{aligned} P(X \geq Y) &= \int_{y=-\infty}^{\infty} \int_{x=y}^{\infty} f_X(x) \cdot f_Y(y) dx dy \\ &= \int_{y=-\infty}^{\infty} f_Y(y) \cdot P(X \geq y) dy \\ &= \int_{y=-\infty}^{\infty} f_Y(y) \cdot P(Y \geq y) dy = 0.5. \end{aligned}$$

Where the equality to 0.5 was proved above.

In our case, $X \succeq Y$. This means that X and Y are independent but are not identically distributed. By definition of stochastic order this also means that $P(X \geq a) > P(Y \geq a)$, for all a with strict inequality for at least one value of a. We get that:

$$\begin{aligned} P(X \geq Y) &= \int_{y=-\infty}^{\infty} \int_{x=y}^{\infty} f_X(x) \cdot f_Y(y) dx dy \\ &= \int_{y=-\infty}^{\infty} f_Y(y) \cdot P(X \geq y) dy \\ &> \int_{y=-\infty}^{\infty} f_Y(y) \cdot P(Y \geq y) dy = 0.5. \end{aligned}$$

Where the last pass holds because X is stochastically larger than Y. We get that $P(X \geq Y) > 0.5$. □

Note that the opposite direction does not always hold, i.e., it is easy to come up with an example where $P(X \geq Y) > 0.5$ but there is no stochastic order between the two random variables. However, the opposite direction is true with an additional assumption that the CDFs do not cross one another (which we do not prove here).

B Hypothesis Testing for Almost Stochastic Dominance

In this section we discuss the implementation of the algorithm for hypothesis testing for the almost stochastic dominance relation between two random variables (empirical score distributions). The code of the algorithm is publicly available.

We are given two sets of scores from two algorithms, n scores from algorithm A and m scores from algorithm B: $A = \{x_1, x_2, ..., x_n\}, B = \{y_1, y_2, ..., y_m\}$. The pseudocode of the algorithm is as follows:

1. Sort the data points from the smallest to the largest in both sets, creating two lists: $A = [x_{(1)}, ..., x_{(n)}]$ and $B = [y_{(1)}, ..., y_{(m)}]$, where $x_{(i)}$ is the i-th smallest value.

2. Build the empirical score distributions F_n, G_m using the following formula:

$$F_n(t) = \frac{1}{n} \sum_{i=1}^{n} 1_{x_{(i)}} \leq t = \frac{\text{number of } x_i\text{s} \leq t}{n}$$

3. Build the empirical inverse score distributions $F^{-1}(t), G^{-1}(t)$ using the following formula:[8]

$$F^{-1}(t) = inf\{x : t \leq F(x)\}, t \in (0, 1)$$

[8] It is possible to compute the inverse CDF without explicitly computing the CDF.

4. Compute the index of stochastic dominance violation $\varepsilon_{\mathcal{W}_2}(F, G)$ (equation 4 of the main paper). In practice we compute the integral operation using the definition of the Riemann integral. That is, when computing $\int_0^1 f(t)dt$ we partition the interval between 0 and 1 into small parts of size Δ and compute the sum of the function value in this part times Δ).

5. Estimate σ: take many samples X_n^*,Y_m^* from the empirical distributions F_n and G_m; for each of those samples compute the expression:

$$\sqrt{\frac{nm}{n+m}}\big(\varepsilon_{\mathcal{W}_2}(F_n^*, G_m^*) - \varepsilon_{\mathcal{W}_2}(F_n, G_m)\big)$$

   and use the variance of those values as the estimate for σ^2, take the square root of that estimator for $\hat{\sigma}_{n,m}$. The more samples, the better. In our implementation we employ the inverse transform sampling method to generate samples.

6. The minimal ϵ for which we can claim that algorithm A is almost stochastically larger than algorithm B with confidence level of $1-\alpha$ is:
   $\epsilon^{min}(F_n, G_m, \alpha) =$
   $\varepsilon_{W_2}(F_n, G_m) - \sqrt{\frac{n+m}{nm}}\hat{\sigma}_{n,m}\Phi^{-1}(\alpha).$