On the Confidence of Neural Network Predictions for some NLP Tasks

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

Neural networks are known to produce unexpected results on inputs that are far from the training distribution. One approach to tackle this problem is to detect the samples on which the trained network can not answer reliably. ODIN is a recently proposed method for out-of-distribution detection that does not modify the trained network and achieves good performance for various image classification tasks. In this paper we adapt ODIN for sentence classification and word tagging tasks. We show that the scores produced by ODIN can be used as a confidence measure for the predictions on both in-distribution and out-of-distribution datasets.

1 Introduction

Neural networks have been shown to perform well on various computer vision and natural language processing tasks. The performance of neural networks is usually measured on the test sets of the corresponding datasets, which does not show how the networks would perform on the samples from other distributions.

Complex neural models that perform well on ImageNet dataset produce nonsensical labels for images far from its training and test sets (see [Nguyen et al., 2015] for examples on unrecognizable images and Figure 1 from [Shafaei et al., 2018] for more realistic images). Similar examples can be found for neural models for NLP tasks. Table 1 shows the results of a simple neural POS tagger on different sentences.

Table 1: The output of a neural POS tagger trained on English-LinES dataset on samples from three different datasets.

UD English-LinES (Accuracy = 14.3%, $\mathrm{PbThreshold}(s) = 0.99598$, $\mathrm{ODIN}(s) = 0.05904$)							
Sentence	Identifying	filters	that	are	currently	in	effect
Ground truth	VERB	NOUN	PRON	VERB	ADV	ADP	NOUN
Predicted labels	ADJ	NOUN	PRON	VERB	ADV	ADP	NOUN
Probabilities	0.97	1.00	1.00	1.00	1.00	1.00	1.00
UD English-EWT (Accuracy = 42.9%, $\mathrm{PbThreshold}(s) = 0.89237$, $\mathrm{ODIN}(s) = 0.05899$)							
Sentence	Try	googling	it	for	more	info	:)
Ground truth	VERB	VERB	PRON	ADP	ADJ	NOUN	SYM
Predicted labels	PRON	VERB	PRON	ADP	ADV	ADV	PUNCT
Probabilities	0.58	1.00	1.00	0.99	1.00	0.70	0.98
UD Dutch-Alpino (Accuracy = 57.1%, $\mathrm{PbThreshold}(s) = 0.99699$, $\mathrm{ODIN}(s) = 0.05906$)							
Sentence	Daarbij	is	een	Macedonische	militair	gedood	.
Ground truth	ADV	AUX	DET	ADJ	NOUN	VERB	PUNCT
Predicted labels	NOUN	AUX	DET	ADJ	ADJ	NOUN	PUNCT
Probabilities	1.00	1.00	0.98	1.00	1.00	1.00	1.00

Submitted to 32nd Conference on Neural Information Processing Systems (NIPS 2018). Do not distribute.

This limitation of neural networks makes it hard to deploy them in critical applications. One of the possible directions to solve the problem is to measure the confidence of the prediction and do not output the prediction if the confidence is low.

Hendrycks and Gimpel [2017] showed that for various neural models, correctly classified examples tend to have higher maximum softmax probabilities than incorrectly classified examples and out-of-distribution examples. They performed several experiments on deep convolutional networks trained on CIFAR-10 and CIFAR-100 and show that by looking at the maximum predicted probability one can classify images from CIFAR-10 and SUN test sets with 95% AUROC.

Liang et al. [2018] improved upon this baseline by using two relatively simple but efficient tricks: (a) by adding temperature to the softmax calculation, (b) by applying adversarial-like perturbation on the inputs. The method is called `ODIN`. Additionally, the authors showed that the performance of `ODIN` strongly depends on the actual distance between the test distributions. For example, if the in-distribution and out-of-distribution datasets are non-intersecting subsets of CIFAR-100, then `ODIN` doesn't work well.

Our contributions are the following:

1. We adapt ODIN for two natural language processing tasks: sentiment analysis and part-of-speech tagging. We show that the two tricks used in `ODIN` are helpful for sentiment analysis, but we could not get improvement from input perturbations for POS tagging.
2. We choose the in-distribution (ID) and out-of-distribution (OOD) datasets in a way that the labels used in the datasets are the same. This choice allows us to measure the accuracy of the model on the test set of the OOD dataset. We show that the scores produced by `ODIN` are relatively higher for the correctly classified examples of the OOD dataset, even when the datasets are close and `ODIN` fails to properly separate them. For part-of-speech tagging, we show that the scores produced by `ODIN` have higher rank correlation with the accuracy of the neural network predictions than the simpler baseline on both ID and OOD datasets.
3. We demonstrate that although character-level embeddings improve POS tagging accuracy, they make it harder to distinguish ID and OOD test sets for the two OOD detection methods. Additionally, the scores produced by these methods have lower rank correlation with the prediction accuracy for all datasets we have tried compared to the models without character-level embeddings.

2 Related Work

There are various ways to approach out-of-distribution detection problem for high dimensional inputs. Shafaei et al. [2018] made a detailed comparison of different approaches on basic image classification tasks. Hendrycks and Gimpel [2017] set a baseline for these methods by looking at the maximum softmax probabilities (`PbThreshold`[1]). Liang et al. [2018] improved upon this baseline using temperature rescaling and perturbations on inputs (`ODIN`). Gal and Ghahramani [2016] showed that Bayesian interpretation of dropout allows to use it to measure the uncertainty of neural network's prediction (`MC-Dropout`). Lakshminarayanan et al. [2017] showed that an ensemble of multiple deep networks can be used to capture uncertainty in a non-Bayesian way. It is important to note that these methods require access to the datasets and to the neural networks trained on them. `PbThreshold` and `ODIN` can be applied to any pretrained neural network. `MC-Dropout` is applicable to any network with a dropout layer (notably, ResNets do not use dropout), and Lakshminarayanan et al. [2017] requires an ensemble of neural networks.

Another class of approaches for OOD detection are based on generative models. Shafaei et al. [2018] evaluated methods based on autoencoders (`AEThreshold`) and `PixelCNN++` [Salimans et al., 2017], and showed that for basic image classification tasks `ODIN` outperforms them (in terms of OOD detection accuracy). Recently, Choi and Jang [2018] used an ensemble of generative models to beat `ODIN` on OOD detection for MNIST and CIFAR-10 datasets (in terms of AUROC). Note that these generative approaches are model-independent, they do not depend on the neural network that performs the actual classification.

[1] We use the codenames of the algorithms from [Shafaei et al., 2018]

Table 2: The datasets used in our experiments along with the performance of the state-of-the-art models on these datasets.

	Labels	Number of sentences			Accuracy of SOTA models
		Train	Dev.	Test	
Yelp Reviews	5	650000	0	50000	70.02 [Howard and Ruder, 2018]
SST-5	5	8544	1101	2210	54.70 [Peters et al., 2018]
en-EWT	17	12543	2002	2077	95.94 [Lim et al., 2018]
en-LinES	17	2738	912	914	97.06 [Lim et al., 2018]
en-GUM	17	2914	707	769	96.44 [Lim et al., 2018]
nl-Alpino	17	12269	718	596	96.90 [Straka, 2018]

Most of the research on OOD detection is focused on image classification tasks. Notable exceptions are [Hendrycks and Gimpel, 2017] and [Shalev et al., 2018]. Applications of more advanced methods to NLP tasks remain largely unexplored.

Additionally, all experiments described above are performed on ID and OOD datasets with different sets of labels. These experiments are good enough to measure the performance of OOD detection. On the other hand, when the distributions of the two datasets have a significant overlap, OOD detection methods fail to produce high accuracy scores. This phenomenon is demonstrated in Section 4.5 of [Liang et al., 2018]. In the context of measuring the confidence of neural network predictions it is fine to misclassify OOD samples as ID unless the network does not produce incorrect answers for the misclassified examples. In order to measure the accuracy of neural models on misclassified OOD samples we choose the ID and OOD datasets to have the same set of labels.

3 Experiments

3.1 Datasets

We performed experiments for two tasks: sentiment analysis and part-of-speech tagging. For sentiment analysis we used the five-class Yelp reviews dataset from [Zhang et al., 2015] and Stanford Sentiment Treebank (SST) [Socher et al., 2013]. SST has labels for every sentence and for every phrase produced from the dependency trees of the sentences. The labels are real numbers between 0 and 1. We created a five-class version of the labels by splitting [0,1] into five equal intervals. For part-of-speech tagging we used two English and one more Dutch treebanks from Universal Dependencies v2.2 [Zeman et al., 2018a] used in the CoNLL Shared Task 2018 [Zeman et al., 2018b]. The majority of the sentences in English-LinES treebank are from literature. English-EWT dataset is larger and is more diverse. The datasets are described in Table 2[2].

3.2 Neural models

In contrast to Liang et al. [2018], we did not use state-of-the-art neural networks. Instead, we trained simple recurrent models for both tasks. For simplicity, we did not use pretrained word embeddings. Similar to [Joulin et al., 2017], we used *hashing trick* to map the words to hashes and embed the hashes using D-dimensional vectors. We randomly initialized the embedding matrix and made it trainable.

Let s be a sentence from one of the datasets, and $w_1, \ldots, w_M$ be the words. The embedding of the m-th word of the sentence will be $x_m = W_e hash(w_m)$. We apply bidirectional LSTM on the embeddings: $\overrightarrow{h_m}, \overleftarrow{h_m} = BiLSTM(x_m)$. For sentiment analysis we apply a dense layer on the concatenation of the last states of the two LSTMs: $f_{sc}(s) = W[\overrightarrow{h_M}, \overleftarrow{h_1}] + b$. The loss function is a cross-entropy: $loss(s) = ce(S(f_{sc}(s), 1))$, where $S_i(\mathbf{z}, T) = \frac{exp(z_i/T)}{\sum_{j=1}^{C} exp(z_j/T)}$ is the modified softmax function, T is the temperature scaling parameter, and C is the number of classes.

[2]State-of-the-art results for sentiment analysis datasets are retrieved from `http://nlpprogress.com` on 27.10.2018. UD POS tagging results are from `http://universaldependencies.org/conll18/results-upos.html`.

Table 3: Out-of-distribution detection performance on part-of-speech tagging tasks. Performance is reported in AUROC scores (higher is better). Char. column indicates whether character-level embeddings were used in the model. ϵ, T column lists the best hyperparameters for `ODIN` found using grid search on validation sets. All scores are in percents.

ID	OOD	Char.	Accuracy		`PbThreshold`			`ODIN`			
			ID	OOD	AUROC			ϵ, T	AUROC		
Yelp	SST-5	No	59.6	24.5	79.9			0.011, 2K	**90.68**		
					Mean	Med.	Tok.		Mean	Med.	Tok.
en-LinES	en-EWT	Yes	89.6	75.9	66.8	70.7	57.1	0, 5K	**75.1**	71.3	58.2
en-LinES	en-EWT	No	87.8	73.4	72.4	73.9	57.9	0, 10K	**77.0**	74.0	59.1
en-LinES	nl-Alpino	Yes	89.6	31.7	95.2	97.4	76.5	0, 2K	**99.1**	98.3	79.8
en-LinES	nl-Alpino	No	87.8	28.6	97.5	97.4	77.4	0, 5	**98.2**	97.9	81.7

For POS tagging we apply a dense layer on every hidden state: $f_{st}(w_m) = W([\overrightarrow{h_m}, \overleftarrow{h_m}]) + b$. The loss function is the average of word-level cross entropies $loss(s) = \frac{1}{M}\sum ce(S(f_{st}(w_m), 1))$.

3.3 Out-of-distribution detection method

We use `ODIN` to detect out-of-distribution samples and compare it with the `PbThreshold` baseline. For every sentence s we compute the scores for each of the methods: $\mathrm{PbThreshold}(s) = \max S(f_{sc}(s))$ and $\mathrm{ODIN}(s) = \max S(f_{sc}(\tilde{\mathbf{x}}), T)$, where $\tilde{\mathbf{x}} = \mathbf{x} + \epsilon\,\mathrm{sign}(\nabla_{\mathbf{x}} S_{\hat{y}}(\mathbf{x})))$, where $\hat{y} = \mathrm{argmax}\, S(\mathbf{x}, 1)$. Here ϵ (perturbation magnitude) and T (temperature) are hyperparameters, which are chosen based on the OOD detection performance on the development sets. For POS tagging, the gradient in the `ODIN` score formula is applied to the mean of word-level probability maximums. The final `ODIN` score of the sentence is some aggregate of the word-level `ODIN` scores. We tried mean and median as the aggregate function. Additionally, we tried to do out-of-distribution detection at the level of tokens, which is expected to be more challenging, as the vocabularies of the datasets have a significant overlap.

Hendrycks and Gimpel [2017] and Liang et al. [2018] report multiple scores for OOD detection performance, while Shafaei et al. [2018] reports only accuracy. We follow [Choi and Jang, 2018] and choose AUROC, as it does not require to tune a threshold. We report the scores on the test sets. Appendix H of [Liang et al., 2018] demonstrates that the choice of the OOD distribution is not critical for the hyperparameter tuning.

3.4 Evaluating confidence scores

For every sentence we produce the scores $\mathrm{PbThreshold}(s)$ and $\mathrm{ODIN}(s)$ and attempt to interpret them as confidence scores for the prediction of the neural network. To measure how well these scores can perform as a confidence measure, we calculate Spearman's rank correlation coefficient between the scores and the accuracy numbers.

4 Results and Discussions

Table 3 shows the results for OOD detection and Table 4 shows the rank correlation coefficients for `PbThreshold` and `ODIN` methods.

The role of the temperature scaling and input perturbations All our experiments confirm the observation from [Liang et al., 2018] that temperature scaling improves out-of-distribution detection. The effect of higher temperatures saturates when T reaches thousands (Figure 1). The positive effect of the perturbations on the inputs is visible for sentiment analysis, but not for POS tagging. We have noticed that when we try to perform OOD detection at the level of tokens, small perturbations ($\epsilon = 0.005$) bring tiny improvements to the performance (by less than 1 AUROC percent point). But for sentence level OOD detection we get worse AUROC scores for $\epsilon > 0$ (or even when $\epsilon < 0$) for both mean and median averaging.

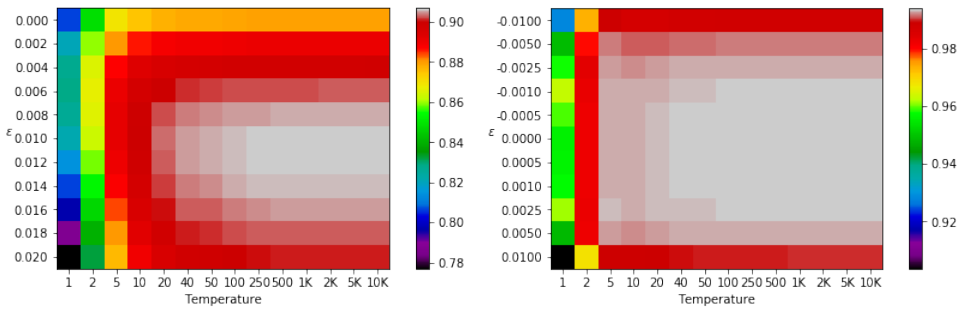

Figure 1: AUROC scores for `ODIN` out-of-distribution detection method for different values of hyperparameters. The left image is for sentiment analysis (Yelp dataset as ID, SST-5 as OOD), the right one is for POS tagging (`English-LinES` as ID, `Dutch-Alpino` as OOD, the neural model uses character-level embeddings).

Sentence-level scores In POS tagging experiments, median of the token-level scores is usually a better score for sentence-level OOD detection than the mean when $T = 1$, while for higher temperatures mean is consistenly better. As expected, token-level OOD detection doesn't work well when both ID and OOD datasets are in English (the vocabularies overlap significantly).

Ranking of the sentences `ODIN` is clearly better than `PbThreshold` according to Spearman's rank correlation coefficient for POS tagging tasks (Table 4). For a neural network trained on `en-LinES`, `ODIN` scores are a good indicator how the network will perform on OOD samples. It is a much better indicator for the closer English dataset than the further Dutch dataset. On the other hand, when we consider the union of ID and OOD datasets, it works better for the union of `en-LinES` and `nl-Alpino` datasets.

To visualize what these correlation coefficients imply in practice, we split the samples into 20 equal buckets according to the scores, and compute accuracy on each of the buckets. We expect to see that the accuracy numbers are monotonically increasing, and that the accuracy numbers on different test sets are close to each other for each bucket. Figure 2 shows that `ODIN` performs slightly better than `PbThreshold`.

For sentiment analysis, we could not get improvement in rank correlation with `ODIN`, although the performance of OOD detection is improved. The reasons of this phenomenon are yet to be investigated.

The role of character-level embeddings Character-level embeddings improve the accuracy of the neural POS tagger for both ID and OOD datasets (consistent with the results reported by Reimers and Gurevych [2017]). On the other hand, they make it harder for `PbThreshold` and `ODIN` methods to separate ID and OOD datasets. Additionally, the scores from the models with character-level embeddings have lower rank correlation with the prediction accuracy. This implies, that the usage of character-level embeddings can be a tradeoff between the accuracy of the model and the reliability of the confidence scores.

Table 4: Spearman's ranking correlation coefficients between sample-level accuracies and the scores produced by `PbThreshold` and `ODIN` (higher is better). For each method we report the coefficient for ID, OOD test sets and the union of both.

ID	OOD	Char.	PbThreshold			ODIN		
			ID	OOD	Both	ID	OOD	Both
en-LinES	nl-EWT	Yes	0.365	0.571	0.565	0.441	0.641	0.636
en-LinES	nl-EWT	No	0.408	0.632	0.616	0.476	0.684	0.676
en-LinES	nl-Alpino	Yes	0.365	0.265	0.793	0.441	0.308	0.828
en-LinES	nl-Alpino	No	0.408	0.312	0.803	0.476	0.445	0.833

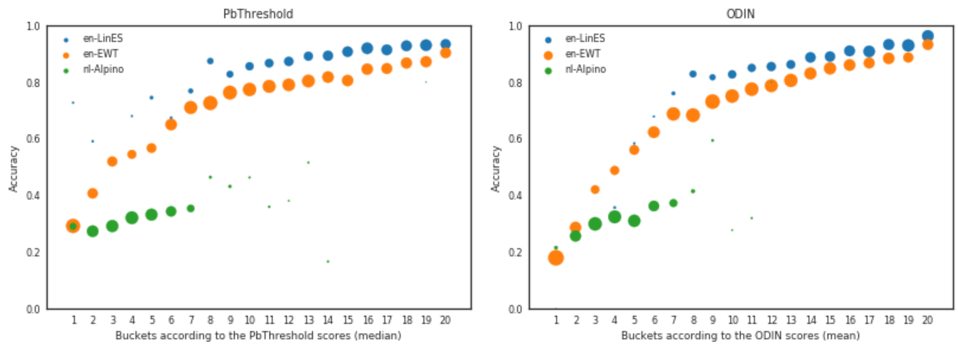

Figure 2: Accuracy of the POS tagger trained on `en-LinES` (without character-level embeddings) on 20 equal buckets of the union of three test sets. The buckets are computed according to the scores produced by each method (`PbThreshold` and `ODIN`). ϵ and T for `ODIN` are determined based on the development sets of `en-LinES` (ID) and `en-EWT` (OOD). The size of a circle is proportional to the number of samples that fall into that bucket. Ideally, accuracy scores for the i-th bucket should be higher than for the $(i-1)$-th bucket, and y coordinates of the three circles for each bucket should be the same.

5 Conclusions and Future Work

In this work we have adapted `ODIN` out-of-distribution detection method on sentence classification and sequence tagging tasks. We showed that as an OOD detector it performs consistently better than for the `PbThreshold` baseline. Additionally, we attempted to quantify how well the scores produced by these methods can be used as confidence scores for the predictions of neural models.

There are many other OOD detection methods that have yet to be tested on NLP tasks. On the other hand, our analysis notably doesn't cover sequence-to-sequence tasks. We have shown that the usage of character-level embeddings makes OOD detection harder for both `PbThreshold` and `ODIN`. The role of the pretrained word vectors, the size of the embeddings, the choice of the neural architecture (recurrent, convolutional or Transformer-like) on OOD detection performance is left for future work.

References

Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 427–436, 2015.

Alireza Shafaei, Mark Schmidt, and James J. Little. Does Your Model Know the Digit 6 Is Not a Cat? A Less Biased Evaluation of Outlier Detectors. *ArXiv e-prints*, 2018.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. 2017. URL `https://openreview.net/forum?id=Hkg4TI9xl`.

Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=H1VGkIxRZ`.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6402–6413, 2017.

Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. In *International*

Conference on Learning Representations, 2017. URL https://openreview.net/forum?id=BJrFC6ceg.

Hyunsun Choi and Eric Jang. Generative ensembles for robust anomaly detection. *arXiv preprint arXiv:1810.01392*, 2018.

Gabi Shalev, Yossi Adi, and Joseph Keshet. Out-of-distribution detection using multiple semantic label representations. *arXiv preprint arXiv:1808.06664*, 2018.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657, 2015.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

Dan Zeman et al. Universal Dependencies 2.2 – CoNLL 2018 shared task development and test data, 2018a. URL http://hdl.handle.net/11234/1-2184. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University, Prague, http://hdl.handle.net/11234/1-2184.

Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–20, Brussels, Belgium, October 2018b. Association for Computational Linguistics. ISBN 978-1-948087-82-7.

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/P18-1031.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

KyungTae Lim, Cheoneum Park, Changki Lee, and Thierry Poibeau. SEx BiST: A multi-source trainable parser with deep contextualized lexical representations. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 143–152, Brussels, Belgium, October 2018. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/K18-2014.

Milan Straka. UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, Brussels, Belgium, October 2018. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/K18-2020.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, volume 2, pages 427–431, 2017.

Nils Reimers and Iryna Gurevych. 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. Association for Computational Linguistics, 2017. doi: 10.18653/v1/D17-1035. URL http://aclweb.org/anthology/D17-1035.