# Handling Noisy Labels for Robustly Learning from Self-Training Data for Low-Resource Sequence Labeling

**Debjit Paul**[*§], **Mittul Singh**[†§], **Michael A. Hedderich**[‡], **Dietrich Klakow**[‡]
[*]Research Training Group AIPHES, Institute for Computational Linguistics,
Heidelberg University, Germany
[†]Department of Signal Processing and Acoustics, Aalto University, Finland
[‡]Spoken Language Systems (LSV), Saarland Informatics Campus,
Saarland University, Germany
paul@cl.uni-heidelberg.de, mittul.singh@aalto.fi,
{mhedderich, dietrich.klakow}@lsv.uni-saarland.de

## Abstract

In this paper, we address the problem of effectively self-training neural networks in a low-resource setting. Self-training is frequently used to automatically increase the amount of training data. However, in a low-resource scenario, it is less effective due to unreliable annotations created using self-labeling of unlabeled data. We propose to combine self-training with noise handling on the self-labeled data. Directly estimating noise on the combined clean training set and self-labeled data can lead to corruption of the clean data and hence, performs worse. Thus, we propose the Clean and Noisy Label Neural Network which trains on clean and noisy self-labeled data simultaneously by explicitly modelling clean and noisy labels separately. In our experiments on Chunking and NER, this approach performs more robustly than the baselines. Complementary to this explicit approach, noise can also be handled implicitly with the help of an auxiliary learning task. To such a complementary approach, our method is more beneficial than other baseline methods and together provides the best performance overall.

## 1 Introduction

For many low-resource languages or domains, only small amounts of labeled data exist. Raw or unlabeled data, on the other hand, is usually available even in these scenarios. Automatic annotation or distant supervision techniques are an option to obtain labels for this raw data, but they often require additional external resources like human-generated lexica which might not be available in a low-resource context. Self-training is a popular technique to automatically label additional text. There, a classifier is trained on a small amount of labeled data and then used to obtain labels for unlabeled instances. However, this can lead to unreliable or noisy labels on the additional data which impede the learning process (Pechenizkiy et al., 2006; Nettleton et al., 2010). In this paper, we focus on overcoming this slowdown of self-training. Hence, we propose to apply noise-reduction techniques during self-training to clean the self-labeled data and learn effectively in a low-resource scenario.

Inspired by the improvements shown by the Noisy Label Neural Network (*NLNN*, Bekker and Goldberger (2016)), we can directly apply *NLNN* to the combined set of the existing clean data and the noisy self-labeled data. However, such an application can be detrimental to the learning process (Section 6). Thus, we introduce the Clean and Noisy Label Neural Network (*CNLNN*) that treats the clean and noisy data separately while training on them simultaneously (Section 3).

This approach leads to two advantages over *NLNN* (Section 6 and 7) when evaluating on two sequence-labeling tasks, Chunking and Named Entity Recognition. **Firstly**, when adding noisy data, *CNLNN* is robust showing consistent improvements over the regular neural network, whereas *NLNN* can lead to degradation in performance. **Secondly**, when combining with an indirect-noise handling technique, i.e. with an auxiliary target in a multi-task fashion, *CNLNN* complements better than *NLNN* in the multi-task setup and overall leads to the best performance.

## 2 Related Work

Self-training has been applied to various NLP tasks, e.g. Steedman et al. (2003) and Sagae and Tsujii (2007). While McClosky et al. (2006) are able to leverage self-training for parsing, Charniak (1997) and Clark et al. (2003) obtain only minimal improvements at best on parsing and POS-tagging

[§]This work was started while the authors were at Saarland University.



*Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 29–34
Minneapolis, Minnesota, June 3 - 5, 2019. 

respectively. In some cases, the results even deteriorate. Other successful approaches of automatically labeling data include using a different classifier trained on out-of-domain data (Petrov et al., 2010) or leveraging external knowledge (Dembowski et al., 2017).

A detailed review of learning in the presence of noisy labels is given in (Frénay and Verleysen, 2014). Recently, several approaches have been proposed for modeling the noise using a confusion matrix in a neural network context. Many works assume that all the data is noisy-labeled (Bekker and Goldberger, 2016; Goldberger and Ben-Reuven, 2017; Sukhbaatar et al., 2015). Hedderich and Klakow (2018) and Hendrycks et al. (2018) propose a setting where a mix of clean and unlabeled data is used. However, they require external knowledge sources for labeling the data or evaluate on synthetic noise. Alternatively, instances with incorrect labels might be filtered out, e.g. in the work by Guan et al. (2011) or Han et al. (2018), but this involves the risk of also filtering out difficult but correct instances. Another orthogonal approach is the use of noise-robust loss functions (Zhang and Sabuncu, 2018).

## 3 Clean and Noisy Label Neural Network

The Noisy Label Neural Network (*NLNN*, Bekker and Goldberger (2016)) assumes that all observed labels in the training set pass through a noise channel flipping some of them from a correct to an incorrect label (see left part of Figure 1). In our scenario, this means that both the human-annotated and the additional automatically-labeled (self-training) corpora are assumed to be noisy. In our experiments (Section 6 and 7), treating both corpora in this fashion degrades the overall performance. To remedy this effect, we propose to treat the human-annotated data as clean data and the self-training data as noisy.

We assume a similar setup as Bekker and Goldberger (2016), training a multi-class neural network soft-max classifier

$$p(y = i|x; w) = \frac{exp(u_i^T h)}{\sum_{j=1}^{k} exp(u_j^T h)}$$

where $x$ is the feature vector, $y$ is the label, $w$ denotes the network weights, $k$ is the number of possible labels, $u$ are soft-max weights and $h = h(x)$ denotes the multi-layer neural network applied to $x$. In contrast to Bekker and Goldberger (2016), we assume that not all of the training data passes

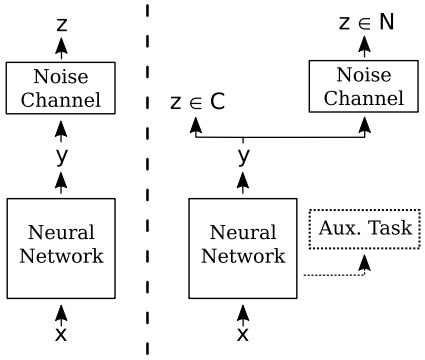


Figure 1: A representation of *NLNN* (left) compared to our proposed *CNLNN* model. The complementary multi-task component (aux. task) is dashed.

through a noisy channel changing the correct labels $y$ to noisy ones ($z \in N$). A part of the training set remains clean ($z \in C$) such that $|C| + |N| = n$ where $n$ is the total number of training examples. The clean labels are a copy of the corresponding correct labels. A schematic representation of this model is shown on the right side of Figure 1. The correct labels $y$ and the noise distribution $\theta$ are hidden for the noisy labels.

We define the probability of observing a label $z$, which can either be noisy or clean and is, thus, dependent on the label's membership to $C$ or $N$:

$$p(z = j|x, w, \theta) = \begin{cases} \sum_{i=1}^{k} p(z = j|y = i; \theta) p(y = i|x; w) & \text{if } z \in N \\ p(y = j|x; w) & \text{if } z \in C \text{ i.e. } z = y \end{cases}$$

Using this probability function and $t$ to index training instances, the log-likelihood of the model parameters is defined as

$$L(w, \theta) = \sum_{z_t \in C} \log p(z_t|x_t, w) + \sum_{z_t \in N} \log(\sum_{i=1}^{k} (p(z_t|y_t = i; \theta) \cdot p(y_t = i|x_t; w))$$

As in Bekker and Goldberger (2016) the model parameters are computed using Expectation Maximization. In the E-step, $\theta$ and $w$ are fixed and an estimate $c$ of the true labels $y$ is obtained for the noisy labels $z$:

$$c_{ti} = p(y_t = i|x_t, z_t; w, \theta) = \frac{p(z_t|y_t = i, \theta) p(y_t = i|x_t; w)}{\sum_j p(z_t|y_t = j; \theta) p(y_t = j|x_t, w)} \quad \text{for } z_t \in N$$

Note that the estimate $c$ is calculated only for the noisy labels whereas the clean labels remain unchanged. Similarly, the noise distribution $\theta$ is calculated only for the noisy labels. The initialization of $\theta$ and the $\theta$'s update step in M-step remain the same as in Bekker and Goldberger (2016), also

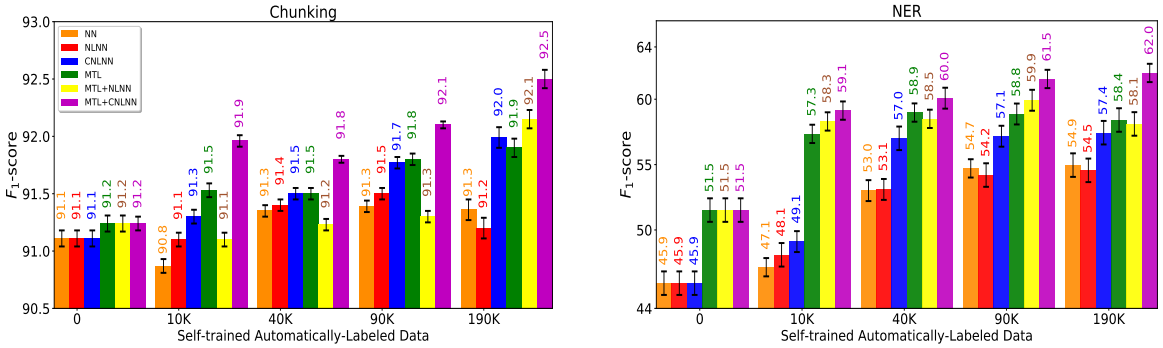


Figure 2: Micro-averaged $F_1$-scores (averaged over five runs) on English Penn Treebank's Chunking and English CoNLL 2003's NER tasks of models from Section 5 are plotted (with error bars) against the amount of automatically-labeled data. 0 on the $x$-axis represents models trained with only the clean training set (10k tokens).

shown below.

$$\theta(i,j) = \frac{\sum_t c_{ti} 1_{\{z_t=j\}}}{\sum_t c_{ti}} \qquad i,j \in \{1,...,k\}, z_t \in N$$

During the M-step, the neural network weights $w$ are estimated as well. The loss function, however, changes compared to the original approach (Bekker and Goldberger, 2016) to (1) and thus, changing the calculation of the gradient to (2):

$$S(w) = \sum_{z_t \in C} \log p(z_t|x_t, w) + \sum_{z_t \in N} \sum_{i=1}^{k} c_{ti} \log p(y_t = i|x_t; w) \quad (1)$$

$$\frac{\partial S}{\partial u_i} = \sum_{z_t \in C} (1_{\{z_t = i\}} - p(z_t|x_t, w))h(x_t) + \sum_{z_t \in N} (c_{ti} - p(y_t|x_t, w))h(x_t) \quad (2)$$

Interestingly, the gradient calculation (2) is a summation of two parts: one to learn from the clean labels and another to learn from the noisy labels. We refer to this model as the Clean and Noisy Label Neural Network (*CNLNN*).

## 4 Training with Noisy Labels in a Multi-Task Setup

*NLNN* and *CNLNN* form explicit ways of handling noise as the noise distribution is calculated during training. In contrast, we can apply a Deep Multi-Task Learning (MTL) approach (Søgaard and Goldberg, 2016), which, unlike *NLNN* and *CNLNN*, does not estimate the noise directly and thus, is an implicit noise-cleaning approach. The MTL method leverages an auxiliary task that augments the data providing other reliable labels and hence, ignoring noisy labels (Ruder, 2017). In our experiments, we combine the implicit noise handling of Deep MTL with the explicit noise handling of *NLNN* and *CNLNN* to complement each other and obtain a more powerful noise handling model than the individual models. Schematic depiction of combining MTL and *CNLNN* is shown in Figure 1. MTL and *NLNN* can also be combined in a similar way.

## 5 Experimental Setup

We evaluate *CNLNN* and other methods on a Chunking and a Named Entity Recognition (NER) task with $F_1$-score as the metric in each case. For Chunking, we use the same data splits as (Søgaard and Goldberg, 2016) based on the English Penn Treebank dataset (Marcus et al., 1993). For NER, the data splits of the English CoNLL 2003 task are used (Sang and Buchholz, 2000). Note that in our NER setup, we evaluate using *BIO-2* labels, so $F_1$-scores reported below might not be comparable to prior work.

To mimic a low resource setting, we limit each training set to the first 10k tokens. The development sets are randomly chosen sentences from the original training set restricted to 1k tokens. The test sets remain unchanged. For the rest of the training data, the original labels are removed and the words are automatically labeled using the baseline model (*NN* described below). We add variable amounts of this automatically-annotated data for self-training in our experiments.

### 5.1 Models

We apply the following models to the above two tasks: **NN** (the simple baseline) is an architecture with bidirectional LSTMs (Hochreiter and Schmidhuber, 1997). For Chunking, we use three LSTM layers, for NER five. The *NN* model, only

trained on the clean data, is used for automatically labeling the raw data (obtaining the noisy data). **NLNN** combines the *NN* with the original noise channel (Bekker and Goldberger, 2016), training it both on clean and noisy instances. **CNLNN** is our new approach of modeling noise, treating clean and noisy labels separately (section 3).

In contrast to the explicit noise handling of *NLNN* and *CNLNN*, we also apply **MTL** for implicit noise handling. Here, we use *NN* as the base architecture and POS-tagging as an auxiliary task. We hypothesise that this low-level task helps the model to generalise its representation and that the POS-tags are helpful because e.g. many named entities are proper nouns. The auxiliary task is trained jointly with the first LSTM layer of *NN* for Chunking and with the second LSTM layer for NER. In our low-resource setting, we use the first 10k tokens of section 0 of Penn Treebank for the auxiliary POS-tagging task for the MTL (Søgaard and Goldberg, 2016). This data is disjunct from the other datasets.

Additionally, we combine both the explicit and implicit noise handling. In the low-resource setting, in general, such a combination addresses the data scarcity better than the individual models. *NLNN* and *CNLNN* combinations with MTL are labeled as **MTL+NLNN** and **MTL+CNLNN** respectively.

### 5.2 Implementation Details

During training, we minimize the cross entropy loss which sums over the entire sentence. The networks are trained with Stochastic Gradient Descent (SGD). To determine the number of iterations for both the NN model and the EM algorithm we use the development data. All models are trained with word embeddings of dimensionality 64 that are initialized with pre-trained Polyglot embeddings (Al-Rfou et al., 2013). We add Dropout (Srivastava et al., 2014) with p=0.1 in between the word embedding layer and the LSTM.

## 6 Results

In Figure 2, we present the $F_1$ scores of the models introduced in the previous section. We perform experiments on Chunking and NER with various amounts of added, automatically-labeled data. In general, adding additional, noisy data tends to improve the performance for all models. This includes the plain *NN*, showing that this model is somewhat robust to noise. Especially for the Chunking task, the possibility for improvement seems limited for *NN* as the performance converges after adding 40k noisy instances. In the Chunking 10k case, the negative effect of the noisy instances results in a score lower than if no data is added.

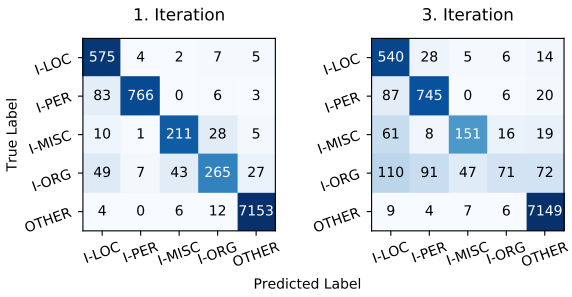


Figure 3: *NLNN* confusion matrices on Chunking's clean training set for 1. and 3. EM iteration. The colors correspond to row-normalized values.

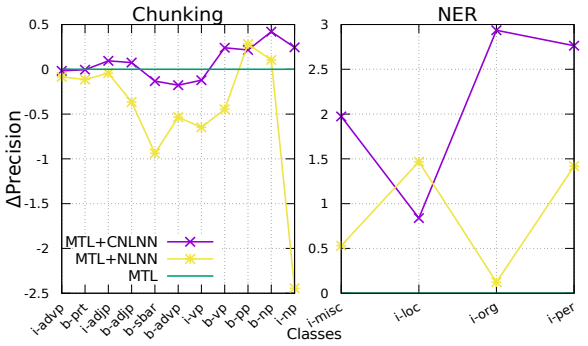


Figure 4: *MTL+CNLNN* vs *MTL+NLNN*: Difference in precision between the combined models and MTL for NER and Chunking test sets with 190K noisy data.

The original *NLNN* model performs similarly to the *NN* model without a noise-handling component. In some cases, the score is even lower. In contrast, *CNLNN* is able to consistently improve over these scores. This demonstrates the importance of our proposed *CNLNN* which treats clean and noisy data separately.

*MTL* is able to improve somewhat over *NN* even without adding automatically-annotated data thanks to the auxiliary task. Additionally, *MTL* performs even better when noisy data is added showing its implicit noise handling capabilities. On their own, both *CNLNN* and *MTL* are able to eliminate some of the negative effects of the noisy data and to leverage the additional data effectively.

Combining *MTL* with *NLNN* results in small improvements at best and can decrease perfor-

mance, especially on Chunking. The best results are achieved with our combined *MTL+CNLNN* model as it outperforms all other models. Even when adding 19 times the amount of self-labeled data, the model is still able to cope with the noise and improve the performance.

## 7 Analysis

**NLNN vs. CNLNN:** In *NLNN*, we observed that clean training tokens were subverted to become noisy in subsequent EM iterations mostly due to the influence of noisy labels from self-labeled data and this effect leads to *NLNN*'s worse performance. Figure 3 presents one such case where the corruption of the confusion matrix from 1. iteration to 3. iteration is displayed. *CNLNN* treats clean and noise data separately and therefore avoids the corruption of clean labels.

**MTL+CNLNN vs. MTL+NLNN:** We noted that *MTL+CNLNN* consistently outperforms *MTL* and *MTL+NLNN*, whereas the *MTL+NLNN* combination can degrade *MTL*'s performance. For nearly all predicted labels the improvements in precision over *MTL* are higher for *MTL+CNLNN* when compared to *MTL+NLNN* (Figure 4). This shows that *CNLNN* complements *MTL* better than *NLNN*.

## 8 Concluding Remarks

In this paper, we apply self-training to neural networks for Chunking and NER in a low-resource setup. Adding automatically-labeled data, the performance of the classifier can wane or can even decline. We propose to mitigate this effect by applying noisy label handling techniques.

However, we found that directly applying an off-the-shelf noise-handling technique as NLNN leads to corruption of the clean training set and worse performance. Thus, we propose the Clean and Noisy Label Neural Network to work separately on the automatically-labeled data. Our model improves the performance faster for a lesser amount of additional data. Moreover, combing the training with auxiliary information can further help handle noise in a complementary fashion.

Meanwhile, more complex neural network architectures (Goldberger and Ben-Reuven, 2017; Luo et al., 2017; Veit et al., 2017) are available for handling noise and we look forward to working with these to upgrade our approach in the future.

## 9 Acknowledgements

This work has been supported by the German Research Foundation as part of the Research Training Group Adaptive Preparation of Information from Heterogeneous Sources (AIPHES) under grant No. GRK 1994/1. We also thank the anonymous reviewers whose comments helped improve this paper.

## References

Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations for multilingual NLP. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning, CoNLL 2013, Sofia, Bulgaria, August 8-9, 2013*, pages 183–192.

Alan Joseph Bekker and Jacob Goldberger. 2016. Training deep neural-networks based on unreliable labels. In *Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2682–2686.

Eugene Charniak. 1997. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Conference on Innovative Applications of Artificial Intelligence*, AAAI'97/IAAI'97, pages 598–603.

Stephen Clark, James R. Curran, and Miles Osborne. 2003. Bootstrapping pos taggers using unlabelled data. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 49–55.

Julia Dembowski, Michael Wiegand, and Dietrich Klakow. 2017. Language independent named entity recognition using distant supervision. In *Proceedings of Language and Technology Conference (LTC)*.

Benoît Frénay and Michel Verleysen. 2014. Classification in the presence of label noise: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5):845–869.

Jacob Goldberger and Ehud Ben-Reuven. 2017. Training deep neural-networks using a noise adaptation layer. In *International Conference on Learning Representations (ICLR)*.

Donghai Guan, Weiwei Yuan, Young-Koo Lee, and Sungyoung Lee. 2011. Identifying mislabeled training data with the aid of unlabeled data. *Applied Intelligence*, 35(3):345–358.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W. Tsang, and Masashi Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels.

In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pages 8536–8546.

Michael A. Hedderich and Dietrich Klakow. 2018. Training a neural network in a low-resource setting on automatically annotated noisy data. In *Proceedings of the Workshop on Deep Learning Approaches for Low-Resource NLP*, pages 12–18. Association for Computational Linguistics.

Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. 2018. Using trusted data to train deep networks on labels corrupted by severe noise. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pages 10477–10486. Curran Associates, Inc.

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

Bingfeng Luo, Yansong Feng, Zheng Wang, Zhanxing Zhu, Songfang Huang, Rui Yan, and Dongyan Zhao. 2017. Learning with noise: Enhance distantly supervised relation extraction with dynamic transition matrix. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Proceedings of the main conference on human language technology conference of the North American Chapter of the Association of Computational Linguistics*, pages 152–159. Association for Computational Linguistics.

David F. Nettleton, Albert Orriols-Puig, and Albert Fornells. 2010. A study of the effect of different types of noise on the precision of supervised learning techniques. *Artificial Intelligence Review*, 33(4):275–306.

Mykola Pechenizkiy, Alexey Tsymbal, Seppo Puuronen, and Oleksandr Pechenizkiy. 2006. Class noise and supervised learning in medical domains: The effect of feature extraction. In *19th IEEE Symposium on Computer-Based Medical Systems*, pages 708–713.

Slav Petrov, Pi-Chuan Chang, Michael Ringgaard, and Hiyan Alshawi. 2010. Uptraining for accurate deterministic question parsing. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 705–713.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv e-prints*, page arXiv:1706.05098.

Kenji Sagae and Jun'ichi Tsujii. 2007. Dependency parsing and domain adaptation with lr models and parser ensembles. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*.

Erik F. Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the conll-2000 shared task: Chunking. In *Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning-Volume 7*, pages 127–132. Association for Computational Linguistics.

Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*.

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

Mark Steedman, Miles Osborne, Anoop Sarkar, Stephen Clark, Rebecca Hwa, Julia Hockenmaier, Paul Ruhlen, Steven Baker, and Jeremiah Crim. 2003. Bootstrapping statistical parsers from small datasets. In *Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics - Volume 1*, EACL '03, pages 331–338.

Sainbayar Sukhbaatar, Joan Bruna, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. 2015. Learning from noisy labels with deep neural networks. In *ICLR Workshop track*.

Andreas Veit, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge Belongie. 2017. Learning from noisy large-scale datasets with minimal supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 839–847.

Zhilu Zhang and Mert R. Sabuncu. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pages 8792–8802.