Exploration of Noise Strategies in Semi-supervised Named Entity Classification

Pooja Lakshmi Narayan
University of Arizona
poojal@email.arizona.edu

Ajay Nagesh[*]
DiDi AI Labs
ajaynagesh@didiglobal.com

Mihai Surdeanu
University of Arizona
msurdeanu@email.arizona.edu

Abstract

Noise is inherent in real world datasets and modeling noise is critical during training as it is effective in regularization. Recently, novel semi-supervised deep learning techniques have demonstrated tremendous potential when learning with very limited labeled training data in image processing tasks. A critical aspect of these semi-supervised learning techniques is augmenting the input or the network with noise to be able to learn robust models. While modeling noise is relatively straightforward in continuous domains such as image classification, it is not immediately apparent how noise can be modeled in discrete domains such as language. Our work aims to address this gap by exploring different noise strategies for the semi-supervised named entity classification task, including statistical methods such as adding Gaussian noise to input embeddings, and linguistically-inspired ones such as dropping words and replacing words with their synonyms. We compare their performance on two benchmark datasets (OntoNotes and CoNLL) for named entity classification. Our results indicate that noise strategies that are linguistically informed perform at least as well as statistical approaches, while being simpler and requiring minimal tuning.

1 Introduction

Modeling noise is a fundamental aspect of machine learning systems. The real world where these systems are deployed are certainly exposed to noisy data. Furthermore, noise is used as an effective regularizer during the training of neural networks (*e.g.*, dropout (Srivastava et al., 2014)). Correct prediction in the presence of noisy input demonstrates robustness of learning systems. A simple analogy to illustrate this is, during image classification, the addition of limited random Gaussian noise to an image can be barely perceived by our visual system and does not drastically change the label a human assigns to an image (Raj, 2018). With the emphasis on compliance and recent advances in adversarial techniques, modeling noise has assumed renewed importance (Goodfellow et al., 2014).

Noise is an important factor in recent state-of-the-art semi-supervised learning systems for image classification (Tarvainen and Valpola, 2017; Rasmus et al., 2015; Miyato et al., 2018). In image processing modeling random noise is relatively straightforward as it is a continuous domain. For instance, adding a small amount random Gaussian jitter can be considered as noisy input. So are other image transformations such as translation, rotation, removing color and so on. However, a discrete domain such as language is not easily amenable to noise augmentation. While one can certainly add random Gaussian noise to embeddings of words (continuous vector representation such as *word2vec* rather than one-hot encoding), the intuition behind such perturbation is not apparent. Algorithms which require explicit modeling of noise require careful thinking in the language domain and is challenging (Clark et al., 2018; Nagesh and Surdeanu, 2018a).

To the best of our knowledge, previous work in the area of modeling noise in natural language processing (NLP) applications has been limited. Clark et al. (2018) acknowledge the difficulty of modeling noise for language and incorporate a simple word dropout in their experiments. So does the work by Nagesh and Surdeanu (2018a). Nagesh and Surdeanu (2018b) add a standard Gaussian perturbation with a fixed variance to the pre-trained word vectors to simulate noise. Belinkov and Bisk (2017) is perhaps one of the most comprehensive works that explore various noise strategies with a different end goal in mind. Their work

[*] work done during AN's post-doc at Univ. of Arizona

*Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 186–191
Minneapolis, June 6–7, 2019. ©2019 Association for Computational Linguistics

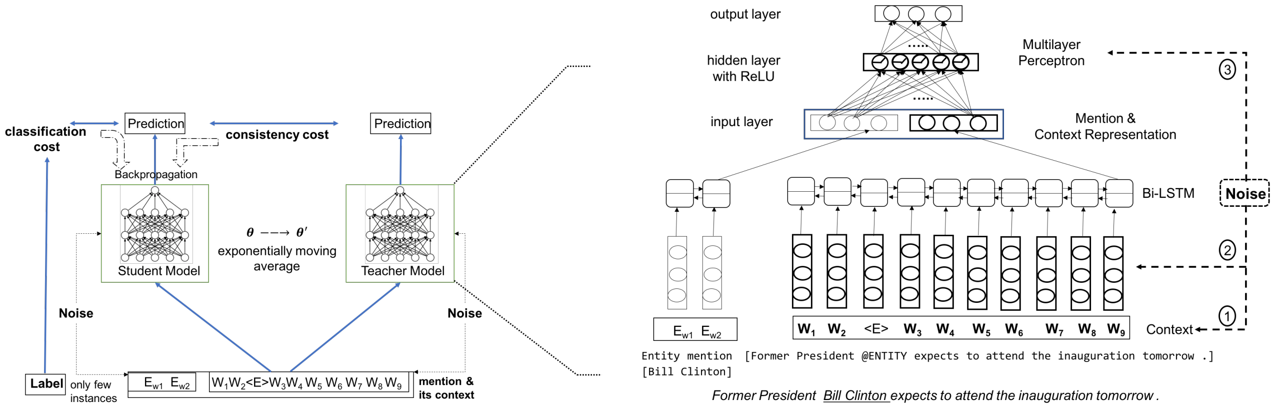
classification cost
Prediction
consistency cost
Prediction
Backpropagation
Student Model
$\theta \dashrightarrow \theta'$
exponentially moving average
Teacher Model
Noise
Noise
Label
only few instances
E_{w1} E_{w2}
W_1W_2<E>$W_3W_4W_5W_6W_7W_8W_9$
mention & its context
output layer
hidden layer with ReLU
input layer
Multilayer Perceptron
Mention & Context Representation
Bi-LSTM
Noise
3
2
1
E_{w1} E_{w2}
W_1 W_2 <E> W_3 W_4 W_5 W_6 W_7 W_8 W_9
Context
Entity mention [Bill Clinton]
[Former President @ENTITY expects to attend the inauguration tomorrow .]
Former President Bill Clinton expects to attend the inauguration tomorrow .

Figure 1: Mean Teacher framework for the named entity classification task (left). E_{wi} are words in the entity mention, W_i are words in the context with entity mention replaced by <E> token. *cost = (classification_cost) +* λ*(consistency_cost).* Unlabeled examples have only consistency cost. Backprop only through student model, teacher model parameters are averaged. The architecture of the student or teacher model (right). Noise can be added to parts in boldface. *predictions = softmax(output_layer)*

explores the degree of robustness of various neural network approaches to different types of noise on a machine translation task.

In this paper, we discuss several noise strategies for the semi-supervised named entity classification task. Some of these, such as word-dropout and synonym-replace, are linguistic and are discrete in nature while others such as Gaussian perturbation to word embeddings are statistical. We show that linguistic noise, while being simple, perform as well as statistical noise. A combination of linguistic and network dropout provides the best performance.

2 Semi-supervised Deep Learning

Semi-supervised learning (SSL) is one of the cornerstones in machine learning (ML) (Zhu, 2005). This is especially true in the case of natural language processing (NLP), as obtaining labeled training data is a costly and tedious process for most of the data-hungry deep learning models.

There has been a flurry of recent work in SSL in the image processing community (Tarvainen and Valpola, 2017; Rasmus et al., 2015). Some of these recent works have achieved impressive performance on hard perceptive tasks. However, repurposing these works to NLP is not a straight forward exercise. As stated earlier, many of these approaches require noise (along with an optional input augmentation step such as rotation) to change the percept slightly, to achieve robust performance. However, augmenting data with noise for NLP tasks is not very clear, as the input domain consists of discrete tokens rather than continuous inputs such as images.

In our previous work (Nagesh and Surdeanu, 2018a), we evaluated three different semi-supervised learning paradigms, namely, bootstrapping-based approaches (Gupta and Manning, 2015), ladder networks (Rasmus et al., 2015) and mean-teacher (Tarvainen and Valpola, 2017) for the semi-supervised named entity classification (NEC) task. The mean-teacher (MT) approach produced the best performance. However, our exploration of noise was limited in the previous study and hence is the focus of the current paper.

The MT framework belongs to the general class of teacher-student networks that learns in the semi-supervised setting *i.e.*, limited supervision and a large amount of unlabeled data and is illustrated in the left part of Figure 1. It consists of two models, termed *student* and *teacher* which are structurally identical but differ in the way their parameters are updated. While the student is updated using regular back-propagation, the parameters of the teacher are a weighted average of the student parameters across different epochs. Further, the cost function is a linear combination of supervision cost (from the limited number amount of supervision) and consistency cost (agreement between the representation from the teacher and student models measured as the L^2 norm difference between them). The motivation of using consistency in the cost function and averaging the parameters in the teacher is to reduce confirmation bias in the teacher when its own predictions are used as pseudo-labels during the training process (akin to averaged perceptron). This provides a strong proxy for the student to rely on in the absence of labeled training data (Tarvainen and Valpola, 2017).

The specific model we employ for semi-supervised named entity classification (NEC) task along with a canonical input data point is depicted in the right part of Figure 1. The input consists of an entity mention and the sentence it appears in, as the context. The goal is to predict the label of the entity. In the semi-supervised setting only a few labeled data points are provided, the rest of the data is unlabeled. We initialize the words in the example with pre-trained word embeddings and run a bi-directional LSTM on both the entity mention and its context. We concatenate the final LSTM state of both the mention and the context representations and run a multi-layer perceptron with one hidden layer to produce the output layer.

A key aspect of the MT framework is the augmentation of the input and/or the network with noise as shown in the right part of Figure 1. We explain this in detail in the next section.

3 Exploration of Noise Strategies

A critical component in the algorithm is the addition of noise to the models. Noise can be added mainly in three key places to the model presented in the previous section as depicted in Figure 1 (parts in boldface). We add a similar but distinct noise to both the teacher and the student models. ① *Input noise* – In the form of linguistically motivated noise such as *word dropout*, or *replacing words* with their synonyms (more details below). ② *Statistical noise* – In the form of standard Gaussian perturbations to pre-trained word embeddings. ③ *Network noise* – Dropout in the intermediate layers of the student and teacher networks.

The idea of adding noise is to regularize the model parameters and help learn robust models in the scenario of very limited labeled training data using the teacher and student models via the consistency cost. Consequently, the MT framework can also be perceived as a consistency regularization technique.

The input noise is applied to the context of an entity mention. The noise was added to a fixed number of words in a context. We explored different types of *input noise*: (1) *Word-dropout* - dropping words randomly in the input context (2) *Synonym-replace* - replace a randomly chosen word in the context by its synonym from WordNet (3) *Word-dropout-idf* - drop the most informative word in the context, as determined by the inverse document frequency (IDF) of context words computed offline. (4) *Synonym-replace-idf* - replace the words in the context according to their IDF (as described above).

For the *statistical noise*, we perturbed the pre-trained word embeddings with standard Gaussian noise with a fixed standard deviation. We varied the amount of standard deviation and the number of words to which this type of noise is added. As we demonstrate in the experiments, this requires careful tuning. Further, adding Gaussian noise is a computationally intensive process as we need to perform this operation in every minibatch during the training process.

We implemented network noise with dropout with fixed probability in both the context representation and the hidden layer in the multi-layer perceptron.

Finally, we combined network noise with input noise. Empirically, we show that this combination yields the best possible performance for the task addressed.

4 Experiments

Task and datasets: The task investigated in this work is named entity classification (NEC), defined as identifying the correct type of an entity mention in a given context, e.g., classifying "*Bill Clinton*" in the sentence "*Former President Bill Clinton expects to attend the inauguration tomorrow.*" as a person name. We define the context as the complete sentence in which the entity mention appears. We use standard benchmark datasets, namely, CoNLL-2003 shared task dataset (Tjong Kim Sang and De Meulder, 2003) and Ontonotes-2013 (Pradhan et al., 2013). Our setting is semi-supervised NEC, so we randomly select a very small percentage of the training dataset (40 datapoints *i.e.* 0.18% of CoNLL and 440 datapoints *i.e.* 0.56% of Ontonotes as labeled data, and artificially remove the labels of the remaining datapoints to simulate the semi-supervised setting. Our task is to predict the correct labels of the unlabeled datapoints. CoNLL had 4 label categories while Ontonotes has 11. We measure the accuracy as the percentage of the datapoints which have

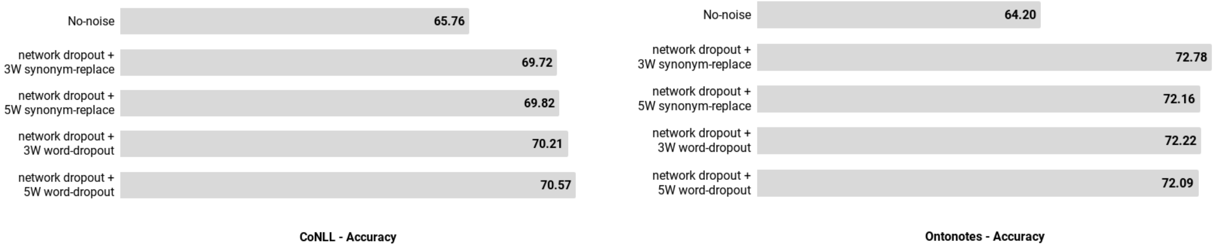

Figure 2: Performance upon combining noise strategies, CoNLL (left) and Ontonotes (right). Best performance: *network dropout + 5W word-dropout* - 70.57% (CoNLL), *network dropout + 3W synonym-replace* - 72.78% (Ontonotes)

		CoNLL	Ontonotes
No noise		65.76 (±2.06)	64.20 (±2.27)
Word-dropout	1 W	67.70 (±2.97)	67.46 (±3.53)
	2 W	68.15 (±3.15)	68.19 (±3.35)
	3 W	**68.54** (±3.38)	68.42 (±3.94)
Synonym-replace	1 W	67.56 (±3.04)	67.70 (±3.20)
	2 W	67.95 (±3.17)	68.40 (±3.62)
	3 W	68.35 (±3.07)	**68.46** (±4.06)
Word-droput-idf	1 W	67.59 (±3.03)	67.38 (±3.29)
	2 W	68.11 (±3.17)	68.14 (±3.63)
	3 W	68.49 (±3.27)	68.30 (±3.77)
Synonym-replace-idf	1 W	67.51 (±3.02)	67.24 (±3.55)
	2 W	67.79 (±3.15)	68.23 (±3.42)
	3 W	68.26 (±3.05)	67.95 (±3.96)
Gaussian (stdev=4)	all W	62.98 (±2.66)	64.89 (±5.12)
Network Dropout		**68.40** (±3.11)	**71.77** (±2.18)

Table 1: Overall accuracies comparing all noise strategies on CoNLL and Ontonotes datasets. *No noise* is the baseline. X W ⇒ X words perturbed by noise. Accuracy is % of correctly classified datapoints. $(\pm y)$ ⇒ variance of 5 runs.

	CoNLL	Ontonotes
1 W	69.70 (±2.93)	68.75 (±3.02)
5 W	68.48 (±2.65)	68.22 (±3.45)
10 W	66.55 (±4.20)	67.32 (±3.42)
stdev=0.05	68.51 (±3.13)	68.42 (±4.15)
stdev=1	66.94 (±2.59)	66.79 (±3.67)
stdev=2	65.43 (±2.68)	65.90 (±4.35)
stdev=4	62.98 (±2.66)	64.94 (±5.91)
stdev=8	62.49 (±2.76)	64.02 (±4.92)
stdev=16	62.87 (±3.08)	64.85 (±6.25)

Table 2: Tuning Gaussian noise - #words & stdev

been predicted with the correct labels.

Experimental settings: We use the entity boundaries for all datapoints during training but only use labels for a small portion of the data as indicated above. We demonstrate an input to our model in the bottom-right of Figure 1. To reduce computational overhead, we filtered out entity mentions which were greater than length 5 from the Ontonotes dataset (4 respectively for CoNLL), and contexts which were greater than length 59 or smaller than length 5 (40 and 3 respectively for CoNLL). Following Nagesh and Surdeanu (2018a), we intialized the pre-trained word-embeddings from Levy and Goldberg (2014) (300d). We ran a 100d bi-directional LSTM on both the entity and context representations, concatenated their outputs and fed them to a 300d multi-layer perceptron with ReLU activations. For network dropout we used $p = 0.2$. This is similar to dropout regularization used in deep neural networks but since the dropout layer drops neurons randomly in teacher and student, this acts as noise in the MT framework. We tried a few variations of this model such as augmenting the LSTM with position embeddings, attention and replacing the LSTM with an average model, but did not observe a considerable improvement in performance.

Results: We present our main results in Table 1. An important note is that the results are the accuracy of classification over 21,373 and 78,492 datapoints in CoNLL and Ontonotes respectively, using only a tiny sliver of the labels in these datasets as supervision. Increasing the number of labeled examples as supervision has the expected effect of improvement in performance. However it is often difficult to obtain sufficient number of examples in the real world. The datapoints for supervision are chosen randomly having equal representation in all classes. The analysis of amount of supervision and its effect on accuracy is reported in Nagesh and Surdeanu (2018a). We report the average (along with their variance) of 5 randomized runs in each noise setting. Our baseline is the *no noise* setting, where the input to the student and teacher models are not augmented by noise.

From Table 1, we observe that adding noise is necessary for good performance, as we see that the various noise strategies consistently improve performance over the baseline on both the datasets. Network noise is a crucial factor for good performance. Input noise which are linguistically motivated, such as *word-dropout* and *synonym-replace* perform as well as the statistical noise. More specifically, *word-dropout* of 3 words and *synonym-replace* of 3 words, are the highest performing non-network noise strategies on CoNLL and Ontonotes respectively. *Synonym-replace* is an interesting strategy as we believe it makes the input more interpretable. In the sense that, the word embedding of a synonym word is closer to the actual word in the vector space. As opposed to gaussian embedding noise, which is a random delta noise added to the embedding to perturb it and we are not sure of its orientation in the high dimensional space. Adding *Gaussian* noise to all words results in performance poorer than or close to baseline. [1] Furthermore, *Gaussian* noise requires fine-tuning over the value of stdev and the number of words on which these should be applied which makes this computationally expensive approach (Table 2). The performance on **-*-idf* runs suggest that random word selection is as good or better. This is ideal, since it is simpler and independent of the data distribution. Finally, *network noise* in combination with linguistic input noise provides the best possible performance, as seen in Figure 2. One possible explanation for this could be that ensembling two high performance systems is akin to combining two good signals achieving better overall results.

5 Conclusion and Future Work

The modeling of noise in discrete domains such as language has received limited focus so far, in the language processing community. In this work we explore several noise strategies for the semi-supervised named entity classification task using the mean teacher framework, where noise augmentation is a crucial factor. We show that linguistic noise such as word-dropout and synonym-replace perform as well as statistical noise, while being simpler and easier to tune. A combination of linguistic and network dropout provides the best performance. As part of future work, we wish to explore noise augmentation in other language processing tasks such as fine-grained entity typing.

[1] In Table 1, for *Gaussian* noise, stdev value is chosen randomly as 4. If we have the luxury to tune this parameter then Table 2 noise, gives the best performance at stdev 0.05.

References

Yonatan Belinkov and Yonatan Bisk. 2017. Synthetic and natural noise both break neural machine translation. *CoRR*, abs/1711.02173.

Kevin Clark, Thang Luong, Christopher D. Manning, and Quoc V. Le. 2018. Semi-supervised sequence modeling with cross-view training.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Sonal Gupta and Christopher D. Manning. 2015. Distributed representations of words to guide bootstrapped entity classifiers. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Baltimore, Maryland. Association for Computational Linguistics.

T. Miyato, S. Maeda, S. Ishii, and M. Koyama. 2018. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1.

Ajay Nagesh and Mihai Surdeanu. 2018a. An exploration of three lightly-supervised representation learning approaches for named entity classification. In *COLING*.

Ajay Nagesh and Mihai Surdeanu. 2018b. Keep your bearings: Lightly-supervised information extraction with ladder networks that avoids semantic drift. In *NAACL HLT 2018*.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Bjrkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using ontonotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152, Sofia, Bulgaria. Association for Computational Linguistics.

Bharath Raj. 2018. Data augmentation - how to use deep learning when you have limited data - part 2. `https://bit.ly/2IvKw1l`. Accessed: 2018-12-10.

Antti Rasmus, Harri Valpola, Mikko Honkala, Mathias Berglund, and Tapani Raiko. 2015. Semi-supervised learning with ladder network. *CoRR*, abs/1507.02672.

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.

Antti Tarvainen and Harri Valpola. 2017. Weight-averaged consistency targets improve semi-supervised deep learning results. *CoRR*, abs/1703.01780.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2003*, pages 142–147. Edmonton, Canada.

Xiaojin Zhu. 2005. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison.