Enhance Robustness of Sequence Labelling with Masked Adversarial Training

Luoxin Chen, Xinyue Liu, Weitong Ruan, Jianhua Lu
Amazon Alexa AI
Cambridge, MA, US
{luoxchen,luxnyu,weiton,jianhual}@amazon.com

Abstract

Adversarial training (AT) has shown strong regularization effects on deep learning algorithms by introducing small input perturbations to improve model robustness. In language tasks, adversarial training brings word-level robustness by adding input noise, which is beneficial for text classification. However, it lacks sufficient contextual information enhancement and thus is less useful for sequence labelling tasks such as chunking and named entity recognition (NER). To address this limitation, we propose masked adversarial training (MAT) to improve robustness from contextual information in sequence labelling. MAT masks or replaces some words in the sentence when computing adversarial loss from perturbed inputs and consequently enhances model robustness using more context-level information. In our experiments, our method shows significant improvements on accuracy and robustness of sequence labelling. By further incorporating with ELMo embeddings, our model achieves better or comparable results to state-of-the-art on CoNLL 2000 and 2003 benchmarks using much less parameters.

1 Introduction

Deep neural network (DNN) based methods have shown great success in various natural language processing (NLP) tasks, such as text classification, sentiment analysis, machine translation, and sequence labelling (Miyato et al., 2017; Ma and Hovy, 2016; Peters et al., 2018). However, some studies (Szegedy et al., 2014; Goodfellow et al., 2015) have illustrated that usually DNN models are not robust enough to input noise. Even adding tiny perturbations to the input could lead to a large increase in loss or mislabelling. To avoid such kind of errors and improve model robustness, adversarial training (AT) (Goodfellow et al., 2015) was proposed to force models to give consistent predictions even with perturbations.

Adversarial training is an approach to improve the robustness and generalization of models, by training models on original examples as well as adversarial examples. Unlike adversarial attack, which aims to find the weakness of models, the objective of adversarial training is to improve the model robustness. Consequently, adversarial attack tends to find the worst adversarial examples, while adversarial training favors adversarial examples which can improve the model robustness and accuracy, not necessarily the worst ones.

Despite the success of adversarial training (AT) on text tasks such as text classification (Miyato et al., 2017) and part-of-speech (POS) tagging (Yasunaga et al., 2018), its gains on other sequence labelling tasks, such as named entity recognition and chunking, are not significant (Yasunaga et al., 2018). AT generates adversarial examples by introducing small but worst-case perturbations to the input embeddings. It is equivalent to replacing words with their close neighbors in embedding space. While this idea improves token-level robustness by ensuring small changes in embedding space would not shift model predictions, replacement by far-away or out-of-vocabulary words is not recoverable by AT. Suppose "I went to Massachusetts by car" is one sentence in the training set and "Massachusetts" is labelled as `LOCATION`. If "Massachusetts" is replaced by "Connecticut", AT is likely to handle it since their embeddings should be close. However, if it is replaced by "Kiyomizu-Dera" (a temple in Kyoto, Japan), AT is unlikely to correctly label this word, since their embeddings might be far away due to lack of co-occurrence. But in this sentence, a human is able to reason its label to `LOCATION` easily by recognizing the contextual phrase "went to" or "by car". Consequently, contextual information is crucial to improve model robustness in context level. Yet, robustness upon

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 297–302
November 16 - 20, 2020.

contextual information is not augmented in adversarial training, and thus its gains to sequence labelling tasks are not that significant and can be further improved.

To incorporate context-level robustness into adversarial training, we propose a new approach, named masked adversarial training (MAT). MAT applies word masks or substitutions (details in Method section) when computing loss from adversarial examples, which forces the model to predict the right labels with no word information or wrong information. For example, if "Massachusetts" in the sentence "I went to Massachusetts by car" is masked or replaced by an irrelevant word such as "pineapple", in order to make the right prediction with such noise, the model has to learn more contextual information from "went to" or "by car". Such approach would enhance model robustness since it relies more about contextual information.

We evaluate MAT on two sequence labelling tasks, named entity recognition (NER) and chunking, which did not exhibit great improvement by using conventional adversarial training in previous literature (Yasunaga et al., 2018). In the experiments, we demonstrate that MAT significantly improves performance on top of AT for those two sequence labelling tasks, and can achieve comparable or better performance than state-of-the-art. Also further analysis indicates that MAT improves generalization over unseen words and unseen patterns.

2 Related Work

Some previous work investigated various approaches to improve contextual robustness during training, such as word dropout (Gal and Ghahramani, 2016), cross-view training (Clark et al., 2018) and masked language model for BERT (Devlin et al., 2018). Word dropout (Gal and Ghahramani, 2016) directly drops some words from input and forces the model to make the same predictions, which simulates that situation where model has not seen some words in training. Cross-view training (Clark et al., 2018) takes auxiliary predictions from neighbor LSTM neurons for each direction and forces them to predict the same label as the current neuron. Hence, cross-view training is equivalent to a stricter word dropout which drops all words before/after the current word. Such dropouts could be treated as augmenting the training data with pieces of input. But in most cases, pieces of input are not valid natural language and thus would create a discrepancy between training and test. Masked language model (Devlin et al., 2018) smooths this inconsistency by applying replacement of tokens for some data while masking the rest (equivalent to word dropout). However, the replacement in masked language model is randomly chosen from the full vocabulary, but the substitution in real scenarios follows some distribution (e.g. replacing "Massachusetts" with a location name is more likely than an animal name), which is not considered in masked language model. Hence, we propose a mask mechanism which is similar to masked language model but has a new substitute selection pipeline to address the concern about substitution probability. We apply it on adversarial examples to force the model to learn more contextual information when optimizing the adversarial loss. Our new approach combines word-level and context-level robustness and achieves superior performance in our experiments.

3 Method

3.1 Model Architecture

Our sequence labelling model adapts CNN-LSTM-CRF architecture, which is used across several best sequence labelling models (Ma and Hovy, 2016; Akbik et al., 2018; Peters et al., 2018; Chen et al., 2020), as shown in Figure 1. We apply a CNN layer to extract character embeddings, concatenate its output with word embeddings and optional ELMo embeddings (Peters et al., 2018) as input features, feed the input features into LSTM layers, and decode with a CRF layer.

3.2 Masked Adversarial Training

3.2.1 Adversarial Examples Generation

In this paper, adversarial perturbations are added to word and character embeddings respectively. To prevent vanishing effects of adversarial perturbations, embeddings are normalized as suggested in (Miyato et al., 2017). Denote w and c as normalized word and character embeddings of the input, η is the rest of the input with no intentional perturbations (ELMo in this paper), θ is the parameters of the model, y is a vector of labels for all tokens in the sequence, and $Loss$ is the loss function. Given bounded norms δ_w and δ_c respectively, the worst case perturbations d_w and d_c for w and c are:

$$d_w = \arg \max_{\epsilon, ||\epsilon||_2 \leq \delta_w} Loss(y; w + \epsilon, c, \eta, \hat{\theta}) \quad (1)$$

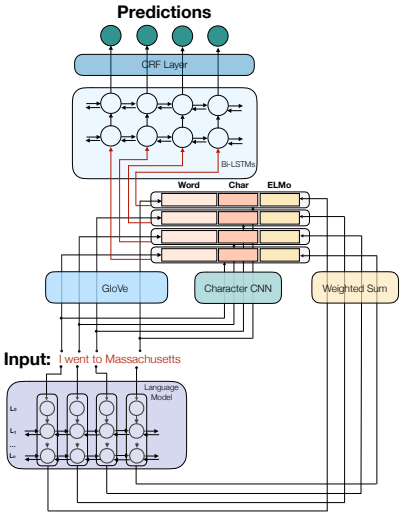

Figure 1: Sequence Labelling Model Architecture

$$d_c = \arg \max_{\tau, ||\tau||_2 \leq \delta_c} Loss(y; w, c + \tau, \eta, \hat{\theta}) \quad (2)$$

Note, w, c and η here are input embeddings for the whole sequence, rather than a single word. Thus, d_w and d_c also represent the sets of worst perturbations for the full sequence, which means we compute the adversarial perturbations for all input words. In addition, $\hat{\theta}$ is current estimation of θ. The purpose of using constant value $\hat{\theta}$ instead of θ is to emphasize that the gradient should not propagate during generation of adversarial examples.

Yet, using equation (1) and (2) to compute exact value of those perturbations with maximization is intractable for complex DNN models. As proposed by Goodfellow et al.(2015), first order approximation is applied to approximate the value of d_w and d_c. With this approximation, d_w and d_c can be calculated by:

$$d_w = \frac{g_w}{||g_w||_2}\delta_w, \quad g_w = \nabla_w Loss(y; w, c, \eta, \hat{\theta}) \quad (3)$$

$$d_c = \frac{g_c}{||g_c||_2}\delta_c, \quad g_c = \nabla_c Loss(y; w, c, \eta, \hat{\theta}) \quad (4)$$

Then, the adversarial loss L_{adv} is formed by:

$$L_{adv} = Loss(y; w + d_w, c + d_c, \eta, \hat{\theta}) \quad (5)$$

3.2.2 Adversarial Examples Masking

The masking process contains three steps. First, some sentences are randomly sampled out, with a rate ρ. Second, one word within each sentence sampled out in previous step will be randomly selected as the candidate for masking or substitution. Finally, candidates will be masked by the word `<Mask>` at a rate of α, or replaced at a rate of $(1 - \alpha)$.

Here, replacing with similar words is more likely to produce valid natural language sentences and thus they should have higher probability during replacement. Consequently, instead of getting a random replacement word from the full vocabulary, a word similarity based replacement is proposed and applied here.

To achieve this, we apply euclidean distance on the un-normalized word embeddings v to measure the word similarity. Then replace the words based on the similarity. Here, we do not want to frequently replace words with their close neighbors since it will have similar effects as AT. In addition, frequent substitution with extremely different words is not ideal either since it is not likely to produce reasonable sentences. As a result, we want to make the probability distribution of sampling substitutes to focus more on the words which share some similarities with the original words but not far-away. For this purpose, we assign a Gaussian function with mean μ and variance Ω^2 to formulate the probability distribution of sampling substitutes based on similarity score. Suppose $s_{i,j}$ is the similarity score between i-th and j-th words, the post score $post_{i,j}$ is:

$$post_{i,j} = Gaussian_{\mu,\Omega^2}(s_{i,j}) \quad (6)$$

Then, we normalize the scores by a softmax and get the probability distribution of replacing words.

3.2.3 Training with Masked Adversarial Loss

Suppose w', c' are embeddings after masking and substitution, the masked adversarial loss L'_{adv} is:

$$L'_{adv} = Loss(y; w' + d_w, c' + d_c, \eta, \hat{\theta}) \quad (7)$$

Here, the conventional perturbations d_w and d_c are still applied to keep the AT effects for words which are not selected for masking/substitution. Also, the size of perturbations is small compared to embeddings. Introducing perturbations to masked/substituted words would not smooth the significant noise from the masking mechanism.

At each training step, the sequence labelling loss is computed as:

$$L_{label} = Loss(y; w, c, \eta, \hat{\theta}) \quad (8)$$

To balance the model accuracy and robustness, a weight λ is introduced to masked adversarial loss L'_{adv}:

$$L_{total} = L_{label} + \lambda L'_{adv} \quad (9)$$

Method	NER	Chunking
Baseline	91.20 ± 0.08	95.18 ± 0.03
Masking (R)	91.30 ± 0.08	95.21 ± 0.04
Masking (S)	$\mathbf{91.35 \pm 0.06}$	$\mathbf{95.24 \pm 0.01}$
AT	91.63 ± 0.07	95.30 ± 0.06
MAT (R)	92.04 ± 0.10	95.42 ± 0.02
MAT (S)	$\mathbf{92.12 \pm 0.07}$	$\mathbf{95.47 \pm 0.03}$

Table 1: Test results on different replacement mechanism. **(R)** stands for random replacement while **(S)** represents similarity based replacement.

This objective function is optimized with respect to θ.

4 Experiments

4.1 Dataset

Our model with masked adversarial training is evaluated on two sequence labelling tasks: named entity recognition and chunking. For named entity recognition, all approaches are evaluated on CoNLL 2003 shared task (Sang and Meulder, 2003). In addition, chunking task is evaluated on the dataset for CoNLL 2000 shared task (Sang and Buchholz, 2000).

4.2 Experiment Settings

While all model parameters are randomly initialized, all the hyper-parameters including initial ELMo weights are chosen by grid search on the development set. ELMo weights are initialized to $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ for NER and $(\frac{1}{5}, \frac{3}{5}, \frac{1}{5})$ for chunking. Variational dropout (Blum et al., 2015) with rate 0.2 is applied to the input and output of each LSTM layer. The selection rate ρ for masked adversarial training is 0.06 and the mask rate α is 0.9. The perturbation sizes for word and character embeddings, δ_w and δ_c, are 0.4 and 0.2 respectively. The weight for masked adversarial loss (i.e. λ) is set to 0.6.

The sequence labelling model is optimized by Adam optimizer (Kingma and Ba, 2015) with batch size 64, learning rate 0.0006 and decay rate 0.992. Early stopping is applied based on model performance on the development set.

5 Evaluation

Two sequence labelling tasks are evaluated with "slot-F1" metric, the same as evaluation metrics in CoNLL 2000 and CoNLL 2003 shared tasks (Sang and Buchholz, 2000; Sang and Meulder, 2003). Considering the relatively small size of the test sets, mean and standard deviation across 5 runs over different random seeds are reported for comparisons.

5.1 Ablation Study

An ablation study on masking and replacement are conducted, and the results are shown in Table 1. For both tasks, word similarity based replacement outperforms random replacement in conditions of baseline and MAT. Considering this consistent benefits of word similarity based replacement, all the experiment results containing MAT are using this replacement mechanism.

5.2 Results

Test results on CoNLL 2000 and 2003 shared task are shown in Table 2. Note that `ELMo-fixed` models consistently perform better than those with trainable ELMo weights in our experiments. So, only baseline models are trained with both fixed and trainable ELMo weights for comparison.

From section 5, 6 and 7, MAT shows significant improvements over AT across all settings. In comparison to previous works, our model outperforms almost all benchmark models in fair comparison setting (with/without additional resources, with/without multi-task training and with/without using development set for training). Only Baevski et al. (2019) reported higher F1-score (93.5 vs 93.48) using a pre-trained CNN model. However, their best model has much more parameters than ours (330M vs 15M). It is valuable to have a much cheaper model with almost same accuracy as state-of-the-art.

For chunking task, our best model (MAT+ELMo-fixed+Multi-task) achieves a new state-of-the-art result (97.04) in CoNLL 2000 benchmark. In addition, MAT consistently beats AT and all other previous benchmark models in the same setting (with/without external resources), even compared to model with larger size (CVT large model has 4 times larger LSTM hidden size than ours).

To further understand the effects of AT and MAT, an additional evaluation on unseen words is performed. Note in this analysis, only models without ELMo are evaluated to get rid of the benefits from ELMo. Token based F1 score is used as the metric for this comparison. As shown in Table 3, while AT improves accuracy on unseen words, MAT gives additional improvement on top of AT in both tasks, which indicates that MAT has better effects on improving model generalization and robustness compared to conventional AT.

Model	NER (F1 ± std)	Chunking (F1 ± std)
AT (Yasunaga et al., 2018)	**91.56**	**95.25**
CVT (Clark et al., 2018)*	92.34 ± 0.06	**96.58 ± 0.04**
BERT-large (Devlin et al., 2018)*	**92.8**	-
ELMo + Multi-Task (Clark et al., 2018)*◇	92.32 ± 0.12	96.83 ± 0.03
CVT+Multi-Task (Clark et al., 2018)*◇	92.42 ± 0.08	96.85 ± 0.05
CVT+Multi-Task+Large (Clark et al., 2018)*◇	**92.61 ± 0.09**	**96.98 ± 0.05**
Flair(Akbik et al., 2018)*†	93.09 ± 0.12	**96.72 ± 0.05**
ELMo+BERT+Flair (Straková et al., 2019)*†	93.38	-
CNN-large + Fine-tune (Baevski et al., 2019)*†	**93.5**	-
Baseline (CNN-LSTM-CRF)	91.20 ± 0.08	95.18 ± 0.03
AT (our implementation)	91.63 ± 0.07	95.30 ± 0.06
MAT	**92.12 ± 0.07**	**95.47 ± 0.03**
Baseline + ELMo*	92.24 ± 0.12	96.49 ± 0.04
Baseline + ELMo-fixed*	92.40 ± 0.10	96.52 ± 0.03
AT + ELMo-fixed*	92.75 ± 0.06	96.63 ± 0.03
MAT + ELMo-fixed*	92.98 ± 0.09	96.94 ± 0.04
MAT + ELMo-fixed + Multi-task*◇	**93.06 ± 0.06**	**97.04 ± 0.02**
Baseline + ELMo-fixed + Dev*†	92.61 ± 0.11	-
AT + ELMo-fixed + Dev*†	93.16 ± 0.07	-
MAT + ELMo-fixed + Dev*†	93.42 ± 0.12	-
MAT + ELMo-fixed + Dev + Multi-task*†◇	**93.48 ± 0.09**	-

Table 2: Test results on CoNLL 2000 and 2003 shared task. The last three sections are our proposed methods. * indicates use of external resources such as pre-trained language model, ◇ represents models jointly trained with other tasks, and † means inclusion of development set into training for NER task. `ELMo-fixed` means using fixed initial weights for ELMo during training, and `Multi-task` indicates that a joint model is trained for all tasks. The best score in each section is marked in **bold**.

Method	NER	Chunking
Baseline	94.29	95.67
AT	94.73	96.45
MAT	**95.12**	**96.86**

Table 3: F1 score on unseen words for two tasks.

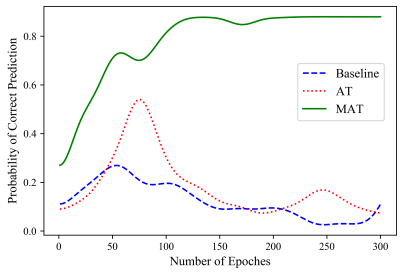

Figure 2: Probability (confidence score) of correctly predicting labels for the sentence "Radio Romania news headlines :". "Romania" only occurs as `LOC` in training set, so the baseline model predicts "Romania" within "Radio Romania news headlines :" as `LOC`. AT shows better chance to label it correctly but fails in the end. MAT gradually learn its contextual information and correctly labels it as `ORG`.

5.3 Robustness Analysis

For the robustness analysis on unseen data, we conduct a case study on the phrase "Radio Romania" whose label is `ORG`, within the sentence "Radio Romania news headlines :", from the CoNLL 2003 dataset. In the training set, "Radio" and "Romania" never show up in the same context. "Radio" only has `ORG` label while "Romania" only has `LOC` label. We draw curves of the probability (confidence score) of correctly labelling this sentence for different models, as shown in Figure 2. Baseline and AT models mislabel it. The probability of correct prediction almost keeps decreasing after some training steps. However, MAT gradually learn its label from contextual information and the probability of right prediction converges to a value larger than 0.8. This case demonstrates the context-level robustness enhancement effects of MAT.

6 Conclusion

In our experiments, we have shown that MAT significantly improves model robustness and generalization on sequence labelling tasks, especially for unseen words or patterns. For the two tasks used in this paper, our approach achieves better or comparable performance to current state-of-the-art with much smaller models. This model architecture is adaptable for all sequence labelling problems and the contextual information brought by MAT has potential benefits for other language tasks.

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 1638–1649.

Alexei Baevski, Sergey Edunov, Yinhan Liu, Luke Zettlemoyer, and Michael Auli. 2019. Cloze-driven pretraining of self-attention networks. *CoRR*, abs/1903.07785.

Avrim Blum, Nika Haghtalab, and Ariel D. Procaccia. 2015. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2575–2583.

Luoxin Chen, Weitong Ruan, Xinyue Liu, and Jianhua Lu. 2020. SeqVAT: Virtual adversarial training for semi-supervised sequence labeling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8801–8811. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Christopher D. Manning, and Quoc V. Le. 2018. Semi-supervised sequence modeling with cross-view training. In *EMNLP 2018, Brussels, Belgium, October 31 - November 4, 2018*, pages 1914–1925.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 1019–1027.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Xuezhe Ma and Eduard H. Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Takeru Miyato, Andrew M. Dai, and Ian J. Goodfellow. 2017. Adversarial training methods for semi-supervised text classification. In *ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237.

Erik F. Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the CoNLL-2000 shared task chunking. In *Fourth Conference on Computational Natural Language Learning, CoNLL 2000, and the Second Learning Language in Logic Workshop, LLL 2000, Held in cooperation with ICGI-2000, Lisbon, Portugal, September 13-14, 2000*, pages 127–132.

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 the Seventh Conference on Natural Language Learning, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, May 31 - June 1, 2003*, pages 142–147.

Jana Straková, Milan Straka, and Jan Hajic. 2019. Neural architectures for nested NER through linearization. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5326–5331.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Michihiro Yasunaga, Jungo Kasai, and Dragomir R. Radev. 2018. Robust multilingual part-of-speech tagging via adversarial training. In *NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 976–986.