Robust Multilingual Part-of-Speech Tagging via Adversarial Training

Michihiro Yasunaga **Jungo Kasai** **Dragomir Radev**
Department of Computer Science, Yale University
{michihiro.yasunaga,jungo.kasai,dragomir.radev}@yale.edu

Abstract

Adversarial training (AT)[1] is a powerful regularization method for neural networks, aiming to achieve robustness to input perturbations. Yet, the specific effects of the robustness obtained from AT are still unclear in the context of natural language processing. In this paper, we propose and analyze a neural POS tagging model that exploits AT. In our experiments on the Penn Treebank WSJ corpus and the Universal Dependencies (UD) dataset (27 languages), we find that AT not only improves the overall tagging accuracy, but also 1) prevents over-fitting well in low resource languages and 2) boosts tagging accuracy for rare / unseen words. We also demonstrate that 3) the improved tagging performance by AT contributes to the downstream task of dependency parsing, and that 4) AT helps the model to learn cleaner word representations. 5) The proposed AT model is generally effective in different sequence labeling tasks. These positive results motivate further use of AT for natural language tasks.

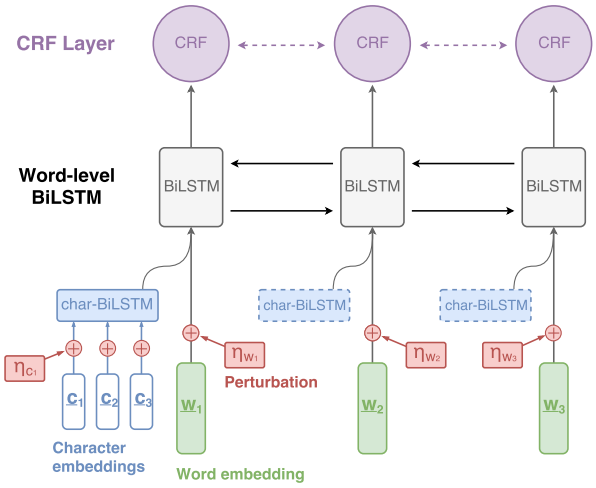

Figure 1: Illustration of our architecture for adversarial POS tagging. Given a sentence, we input the normalized word embeddings ($\underline{\boldsymbol{w}}_1, \underline{\boldsymbol{w}}_2, \underline{\boldsymbol{w}}_3$) and character embeddings (showing $\underline{\boldsymbol{c}}_1, \underline{\boldsymbol{c}}_2, \underline{\boldsymbol{c}}_3$ for w_1). Each word is represented by concatenating its word embedding and its character-level BiLSTM output. They are fed into the main BiLSTM-CRF network for POS tagging. In adversarial training, we compute and add the worst-case perturbation η to all the input embeddings for regularization.

1 Introduction

Recently, neural network-based approaches have become popular in many natural language processing (NLP) tasks including tagging, parsing, and translation (Chen and Manning, 2014; Bahdanau et al., 2015; Ma and Hovy, 2016). However, it has been shown that neural networks tend to be locally unstable and even tiny perturbations to the original inputs can mislead the models (Szegedy et al., 2014). Such maliciously perturbed inputs are called *adversarial examples*. *Adversarial training* (Goodfellow et al., 2015) aims to improve the robustness of a model to input perturbations by training on both unmodified examples and adversarial examples. Previous work (Goodfellow et al., 2015; Shaham et al., 2015) on image recognition has demonstrated the enhanced robustness of their models to unseen images via adversarial training and has provided theoretical explanations of the regularization effects.

Despite its potential as a powerful regularizer, adversarial training (AT) has yet to be explored extensively in natural language tasks. Recently, Miyato et al. (2017) applied AT on text classification, achieving state-of-the-art accuracy. Yet, the specific effects of the robustness obtained from AT are still unclear in the context of NLP. For example, research studies have yet to answer questions such as 1) how can we interpret perturbations or robustness on natural language inputs? 2) how are they related to linguistic factors like vocabulary statis-

[1]We distinguish AT from Generative Adversarial Networks (GANs).

Proceedings of NAACL-HLT 2018, pages 976–986
New Orleans, Louisiana, June 1 - 6, 2018.

tics? 3) are the effects of AT language-dependent? Answering such questions is crucial to understand and motivate the application of adversarial training on natural language tasks.

In this paper, spotlighting a well-studied core problem of NLP, we propose and carefully analyze a neural part-of-speech (POS) tagging model that exploits adversarial training. With a BiLSTM-CRF model (Huang et al., 2015; Ma and Hovy, 2016) as our baseline POS tagger, we apply adversarial training by considering perturbations to input word/character embeddings. In order to demystify the effects of adversarial training in the context of NLP, we conduct POS tagging experiments on multiple languages using the Penn Treebank WSJ corpus (Englsih) and the Universal Dependencies dataset (27 languages), with thorough analyses of the following points:

- Effects on different target languages
- Vocabulary statistics and tagging accuracy
- Influence on downstream tasks
- Representation learning of words

In our experiments, we find that our adversarial training model consistently outperforms the baseline POS tagger, and even achieves state-of-the-art results on 22 languages. Furthermore, our analyses reveal the following insights into adversarial training in the context of NLP:

- The regularization effects of adversarial training (AT) are general across different languages. AT can prevent overfitting especially well when training examples are scarce, providing an effective tool to process low resource languages.
- AT can boost the tagging performance for rare/unseen words and increase the sentence-level accuracy. This positively affects the performance of down-stream tasks such as dependency parsing, where low sentence-level POS accuracy can be a bottleneck (Manning, 2011).
- AT helps the network learn cleaner word embeddings, showing stronger correlations with their POS tags.

We argue that the effects of AT can be interpreted from the perspective of natural language. Finally, we demonstrate that the proposed AT model is generally effective across different sequence labeling tasks. This work therefore provides a strong motivation and basis for utilizing adversarial training in NLP tasks.

2 Related Work

2.1 POS Tagging

Part-of-speech (POS) tagging is a fundamental NLP task that facilitates downstream tasks such as syntactic parsing. While current state-of-the-art POS taggers (Ling et al., 2015; Ma and Hovy, 2016) yield accuracy over 97.5% on PTB-WSJ, there still remain issues. The per token accuracy metric is easy since taggers can easily assign correct POS tags to highly unambiguous tokens, such as punctuation (Manning, 2011). Sentence-level accuracy serves as a more realistic metric for POS taggers but it still remains low. Another problem with current POS taggers is that their accuracy deteriorates drastically on low resource languages and rare words (Plank et al., 2016). In this work, we demonstrate that adversarial training (AT) can mitigate these issues.

It is empirically shown that POS tagging performance can greatly affect downstream tasks such as dependency parsing (Dozat et al., 2017). In this work, we also demonstrate that the improvements obtained from our AT POS tagger actually contribute to dependency parsing. Nonetheless, parsing with gold POS tags still yields better results, bolstering the view that POS tagging is an essential task in NLP that needs further development.

2.2 Adversarial Training

The concept of adversarial training (Szegedy et al., 2014; Goodfellow et al., 2015) was originally introduced in the context of image classification to improve the robustness of a model by training on input images with malicious perturbations. Previous work (Goodfellow et al., 2015; Shaham et al., 2015; Wang et al., 2017) has provided a theoretical framework to understand adversarial examples and the regularization effects of adversarial training (AT) in image recognition.

Recently, Miyato et al. (2017) applied AT to a natural language task (text classification) by extending the concept of adversarial perturbations to word embeddings. Wu et al. (2017) further explored the possibility of AT in relation extraction. Both report improved performance on their tasks via AT, but the specific effects of AT have yet to be analyzed. In our work, we aim to address this issue by providing detailed analyses on the effects of AT from the perspective of NLP, such as different languages, vocabulary statistics, word embedding distribution, and aim to motivate future research

that exploits AT in NLP tasks.

AT is related to other regularization methods that add noise to data such as dropout (Srivastava et al., 2014) and its variant for NLP tasks, word dropout (Iyyer et al., 2015). Xie et al. (2017) discuss various data noising techniques for language modeling. While these methods produce random noise, AT generates perturbations that the current model is particularly vulnerable to, and thus is claimed to be effective (Goodfellow et al., 2015).

It should be noted that while related in name, adversarial training (AT) differs from Generative Adversarial Networks (GANs) (Goodfellow et al., 2014). GANs have already been applied to NLP tasks such as dialogue generation (Li et al., 2017) and transfer learning (Kim et al., 2017; Gui et al., 2017). Adversarial training also differs from adversarial *evaluation*, recently proposed for reading comprehension tasks (Jia and Liang, 2017).

3 Method

In this section, we introduce our baseline POS tagging model and explain how we implement adversarial training on top.

3.1 Baseline POS Tagging Model

Following the recent top-performing models for sequence labeling tasks (Plank et al., 2016; Lample et al., 2016; Ma and Hovy, 2016), we employ a Bi-directional LSTM-CRF model as our baseline (see Figure 1 for an illustration).

Character-level BiLSTM. Prior work has shown that incorporating character-level representations of words can boost POS tagging accuracy by capturing morphological information present in each language. Major neural character-level models include the character-level CNN (Ma and Hovy, 2016) and (Bi)LSTM (Dozat et al., 2017). A Bi-directional LSTM (BiLSTM) (Hochreiter and Schmidhuber, 1997; Schuster and Paliwal, 1997) processes each sequence both forward and backward to capture sequential information, while preventing the vanishing / exploding gradient problem. We observed that the character-level BiLSTM outperformed the CNN by 0.1% on the PTB-WSJ development set, and hence in all of our experiments we use the character-level BiLSTM. Specifically, we generate a character-level representation for each word by feeding its character embeddings into the BiLSTM and obtaining the concatenated final states.

Word-level BiLSTM. Each word in a sentence is represented by concatenating its word embedding and its character-level representation. They are fed into another level of BiLSTM (word-level BiLSTM) to process the entire sentence.

CRF. In sequence labeling tasks it is beneficial to consider the correlations between neighboring labels and jointly decode the best chain of labels for a given sentence. With this motivation, we apply a conditional random field (CRF) (Lafferty et al., 2001) on top of the word-level BiLSTM to perform POS tag inference with global normalization, addressing the "label bias" problem. Specifically, given an input sentence, we pass the output sequence of the word-level BiLSTM to a first-order chain CRF to compute the conditional probability of the target label sequence:

$$p(\boldsymbol{y} \mid \boldsymbol{s}; \boldsymbol{\theta})$$

where $\boldsymbol{\theta}$ represents all of the model parameters (in the BiLSTMs and CRF), $\boldsymbol{s}$ and $\boldsymbol{y}$ denote the input embeddings and the target POS tag sequence, respectively, for the given sentence.

For training, we minimize the negative log-likelihood (loss function)

$$L(\boldsymbol{\theta}; \boldsymbol{s}, \boldsymbol{y}) = -\log p(\boldsymbol{y} \mid \boldsymbol{s}; \boldsymbol{\theta}) \quad (1)$$

with respect to the model parameters. Decoding searches for the POS tag sequence $\boldsymbol{y}^*$ with the highest conditional probability using the Viterbi algorithm. For more detail about the BiLSTM-CRF formulation, refer to Ma and Hovy (2016).

3.2 Adversarial Training

Adversarial training (Goodfellow et al., 2015) is a powerful regularization method, primarily explored in image recognition to improve the robustness of classifiers to input perturbations. Given a classifier, we first generate input examples that are very close to original inputs (so should yield the same labels) yet are likely to be misclassified by the current model. Specifically, these *adversarial examples* are generated by adding small perturbations to the inputs in the direction that significantly increases the loss function of the classifier (*worst-case* perturbations). Then, the classifier is trained on the mixture of clean examples and adversarial examples to improve the stability to input perturbations. In this work, we incorporate adversarial training into our baseline POS tagger, aiming to achieve better regularization effects and to provide their interpretations in the context of NLP.

Generating adversarial examples. Adversarial training (AT) considers continuous perturbations to inputs, so we define perturbations at the level of dense word / character embeddings rather than one-hot vector representations, similarly to Miyato et al. (2017). Specifically, given an input sentence, we consider the concatenation of all the word/character embeddings in the sentence: $\boldsymbol{s} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{c}_1, \boldsymbol{c}_2, \ldots]$. To prepare an adversarial example, we aim to generate the worst-case perturbation of a small bounded norm ϵ that maximizes the loss function L of the current model:

$$\boldsymbol{\eta} = \underset{\boldsymbol{\eta}': \|\boldsymbol{\eta}'\|_2 \leq \epsilon}{\arg\max}\ L(\hat{\boldsymbol{\theta}}; \boldsymbol{s} + \boldsymbol{\eta}', \boldsymbol{y})$$

where $\hat{\boldsymbol{\theta}}$ is the current value of the model parameters, treated as a constant, and $\boldsymbol{y}$ denotes the target labels. Since the exact computation of such $\boldsymbol{\eta}$ is intractable in complex neural networks, we employ the Fast Gradient Method (Liu et al., 2017; Miyato et al., 2017) i.e. first order approximation to obtain an approximate worst-case perturbation of norm ϵ, by a single gradient computation:

$$\boldsymbol{\eta} = \epsilon\, \boldsymbol{g}/\|\boldsymbol{g}\|_2, \text{ where } \boldsymbol{g} = \nabla_{\boldsymbol{s}} L(\hat{\boldsymbol{\theta}}; \boldsymbol{s}, \boldsymbol{y}) \quad (2)$$

ϵ is a hyperparameter to be determined in the development dataset. Note that the perturbation $\boldsymbol{\eta}$ is generated in the direction that significantly increases the loss L. We find such $\boldsymbol{\eta}$ against the current model parameterized by $\hat{\boldsymbol{\theta}}$, at each training step, and construct an adversarial example by

$$\boldsymbol{s}_{\text{adv}} = \boldsymbol{s} + \boldsymbol{\eta}$$

However, if we do not restrict the norm of word / character embeddings, the model could trivially learn embeddings of large norms to make the perturbations insignificant. To prevent this issue, we normalize word/character embeddings so that they have mean 0 and variance 1 for every entry, as in Miyato et al. (2017). The normalization is performed every time we feed input embeddings into the LSTMs and generate adversarial examples. To ensure a fair comparison, we also normalize input embeddings in our baseline model.

While Miyato et al. (2017) set the norm of a perturbation ϵ (Eq 2) to be a fixed value for all input sentences, to generate adversarial examples for an entire sentence of a variable length and to include character embeddings besides word embeddings, we make the perturbation size ϵ adaptive to the dimension of the concatenated input embedding $\boldsymbol{s} \in \mathbb{R}^D$. We set ϵ to be $\alpha\sqrt{D}$ (i.e., proportional to $\sqrt{D}$), as the expected squared norm of $\boldsymbol{s}$ after the embedding normalization is D. The scaling factor α is selected from $\{0.001, 0.005, 0.01, 0.05, 0.1\}$ based on the development performance in each treebank. We used 0.01 for PTB-WSJ and UD-Spanish, and 0.05 for the rest. Note that $\alpha = 0$ would generate no noise (identical to the baseline); if $\alpha = 1$, the generated adversarial perturbation would have a norm comparable to the original embedding, which could change the semantics of the input sentence (Wu et al., 2017). Hence, the optimal perturbation scale α should lie in between and be small enough to preserve the semantics of the original input.

Adversarial training. At each training step, we generate adversarial examples against the current model, and train on the mixture of clean examples and adversarial examples to achieve robustness to input perturbations. To this end, we define the loss function for adversarial training as:

$$\tilde{L} = \gamma L(\boldsymbol{\theta}; \boldsymbol{s}, \boldsymbol{y}) + (1 - \gamma) L(\boldsymbol{\theta}; \boldsymbol{s}_{\text{adv}}, \boldsymbol{y})$$

where $L(\boldsymbol{\theta}; \boldsymbol{s}, \boldsymbol{y})$, $L(\boldsymbol{\theta}; \boldsymbol{s}_{\text{adv}}, \boldsymbol{y})$ represent the loss from a clean example and the loss from its adversarial example, respectively, and γ determines the weighting between them. We used $\gamma = 0.5$ in all our experiments. This objective function can be optimized with respect to the model parameters $\boldsymbol{\theta}$, in the same manner as the baseline model.

4 Experiments

To fully analyze the effects of adversarial training, we train and evaluate our baseline/adversarial POS tagging models on both a standard English dataset and a multilingual dataset.

4.1 Datasets

As a standard English dataset, we use the Wall Street Journal (WSJ) portion of the Penn Treebank (PTB) (Marcus et al., 1993), containing 45 different POS tags. We adopt the standard split: sections 0-18 for training, 19-21 for development and 22-24 for testing (Collins, 2002; Manning, 2011).

For multilingual POS tagging experiments, to compare with prior work, we use treebanks from Universal Dependencies (UD) v1.2 (Nivre et al., 2015) (17 POS) with the given data splits. We experiment on languages for which pre-trained Polyglot word embeddings (Al-Rfou et al., 2013) are available, resulting in 27 languages listed in Table 2. We regard languages with less than 60k tokens of training data as low-resource (Table 2, bottom), as in Plank et al. (2016).

Model	Accuracy
Toutanova et al. (2003)	97.27
Manning (2011)	97.28
Collobert et al. (2011)	97.29
Søgaard (2011)	97.50
Ling et al. (2015)	**97.78**
Ma and Hovy (2016)	97.55
Yang et al. (2017)	97.55
Hashimoto et al. (2017)	97.55
Ours – Baseline (BiLSTM-CRF)	97.54
Ours – Adversarial	97.58

Table 1: POS tagging accuracy on the PTB-WSJ test set, with other top-performing systems.

4.2 Training & Evaluation Details

Model settings. We initialize word embeddings with 100-dimensional GloVe (Pennington et al., 2014) for English, and with 64-dimensional Polyglot (Al-Rfou et al., 2013) for other languages. We use 30-dimensional character embeddings, and set the state sizes of character/word-level BiLSTM to be 50, 200 for English, 50, 100 for low resource languages, and 50, 150 for other languages. The model parameters and character embeddings are randomly initialized, as in Ma and Hovy (2016). We apply dropout (Srivastava et al., 2014) to input embeddings and BiLSTM outputs for both baseline and adversarial training, with dropout rate 0.5.

Optimization. We train the model parameters and word/character embeddings by the mini-batch stochastic gradient descent (SGD) with batch size 10, momentum 0.9, initial learning rate 0.01 and decay rate 0.05. We also use a gradient clipping of 5.0 (Pascanu et al., 2012). The models are trained with early stopping (Caruana et al., 2001) based on the development performance.

Evaluation. We evaluate per token tagging accuracy on test sets. We repeat the experiment three times and report the statistical significance.

4.3 Results

PTB-WSJ dataset. Table 1 shows the POS tagging results. As expected, our baseline (BiLSTM-CRF) model (accuracy 97.54%) performs on par with other state-of-the-art systems. Built upon this baseline, our adversarial training (AT) model reaches accuracy 97.58% thanks to its regularization power, outperforming recent POS taggers except Ling et al. (2015). The improvement over the baseline is statistically significant, with p-value $<$ 0.05 on the t-test. We provide additional analysis on this result in later sections.

	Our Models		Plank et al. (2016)			Berend	Nguyen et
	Baseline	Adversarial	BiLSTM	TNT	CRF	(2017)	al. (2017)
bg	98.34	**98.53**	97.97	96.84	96.36	95.63	97.4
cs	98.70	**98.81**	98.24	96.82	96.56	95.83	–
da	96.63	**96.74**	96.35	94.29	93.83	93.32	95.8
de•	94.29	**94.35**	93.38	92.64	91.38	90.73	92.7
en	95.72	**95.82**	95.16	94.55	93.35	93.47	94.7
es	96.26	**96.44**	95.74	94.55	94.23	94.69	95.9
eu•	94.55	94.71	**95.51**	93.35	91.63	90.63	93.7
fa	97.38	**97.51**	97.49	95.98	95.65	96.11	96.8
fi•	94.54	95.40	**95.85**	93.59	90.32	89.19	94.6
fr	96.48	**96.63**	96.11	94.51	95.14	94.96	96.0
he	97.34	**97.43**	96.96	93.71	93.63	95.28	–
hi	97.12	**97.21**	97.10	94.53	96.00	96.09	96.4
hr•	96.12	96.32	**96.82**	94.06	93.16	93.53	–
id	93.95	**94.03**	93.41	93.16	92.96	92.02	93.1
it	98.04	**98.08**	97.95	96.16	96.43	96.28	97.5
nl	92.64	93.09	**93.30**	88.54	90.03	85.10	91.4
no	97.88	**98.08**	98.03	96.31	96.21	95.67	97.4
pl•	97.34	97.57	**97.62**	95.57	93.96	93.95	96.3
pt	97.94	**98.07**	97.90	96.27	96.32	95.50	97.5
sl•	97.81	**98.11**	96.84	94.92	94.77	92.70	97.1
sv	96.39	**96.70**	96.69	95.19	94.45	94.62	–
Avg	96.45	**96.65**	96.40	94.55	94.11	93.59	95.55
el	98.18	**98.24**	–	–	–	97.12	–
et•	90.79	**91.32**	–	–	–	86.30	–
ga	90.66	**91.11**	–	–	–	88.82	–
hu•	93.39	**94.02**	–	–	–	89.47	–
ro	91.24	**91.46**	–	–	–	88.99	–
ta	82.91	**83.16**	–	–	–	81.80	–
Avg	91.20	**91.55**	–	–	–	88.41	–

Table 2: POS tagging accuracy (test) for 27 UD v1.2 treebanks, with other recent works, Plank et al. (2016), Berend (2017) and Nguyen et al. (2017). For Plank et al. (2016), we include the traditional baselines TNT and CRF, and their state-of-the-art model that employs a multi-task BiLSTM. Languages with • are morphologically rich, and those at the bottom ('el' to 'ta') are low-resource, containing less than 60k tokens in their training sets.

Multilingual dataset (UD). Experimental results are summarized in Table 2. Our AT model shows clear advantages over the baseline in all of the 27 languages (average improvement ∼0.25%; see the two shaded columns). Considering that our baseline (BiLSTM-CRF) is already a top performing model for POS tagging, these improvements made by AT are substantial. The improvements are also statistically significant for all the languages, with p-value $<$ 0.05 on the t-test, suggesting that the regularization by AT is generally effective across different languages. Moreover, our AT model achieves state-of-the-art on nearly all of the languages, except the five where Plank et al. (2016)'s multi-task BiLSTM yielded better results. Among the five, most languages are morphologically rich (•).[2] We suspect that their joint training of word rarity may be of particular help in processing morphologically complex words.

[2]We followed the criteria of morphological richness used in Nguyen et al. (2017).

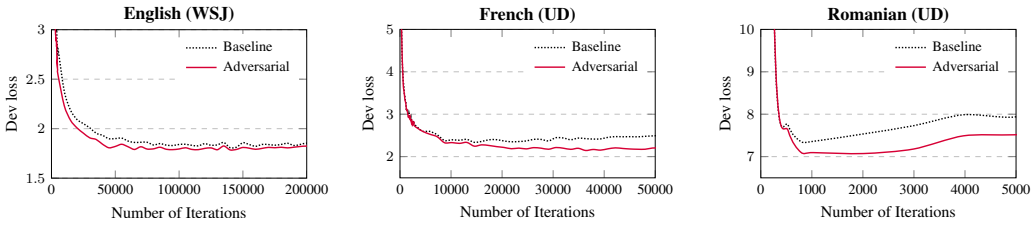

Figure 2: Learning curves for three representative languages (Romanian is low-resource). We show the transition of loss (defined in Eq 1) on the development sets.

English (WSJ)

Word Frequency	0	1-10	10-100	100-	Total
# Tokens	3240	7687	20908	97819	129654
Baseline	**92.25**	95.36	96.03	98.19	97.53
Adversarial	92.01	**95.52**	96.10	98.23	97.57

French (UD)

Word Frequency	0	1-10	10-100	100-	Total
# Tokens	356	839	1492	4523	7210
Baseline	87.64	94.05	94.03	98.43	96.48
Adversarial	87.92	**94.88**	94.03	98.50	96.63

Table 3: POS tagging accuracy (test) on different subsets of words, categorized by their frequency of occurrence in training. The second row shows the number of tokens in the test set that are in each category. The third and fourth rows show the performance of our two models. Better scores are underlined. The biggest improvement is **in bold**.

English (WSJ)

Word Frequency	0	1-10	10-100	100-	Total
# Tokens	6480	15374	41815	195637	259306
Baseline	97.76	97.71	97.80	97.45	97.53
Adversarial	**98.06**	97.71	97.89	97.47	97.57

French (UD)

Word Frequency	0	1-10	10-100	100-	Total
# Tokens	712	1678	2983	9045	14418
Baseline	95.08	97.08	97.58	96.11	96.48
Adversarial	**95.37**	97.26	97.79	96.23	96.63

Table 4: POS tagging accuracy (test) on *neighboring* words. We cluster all words in the test set in the same way as Table 3 and consider the tagging performance on the neighbors (left and right) of these words in the test text.

Additionally, we see that our AT model achieves notably large improvements over the baseline in resource-poor languages (the bottom of Table 2), with average improvement 0.35%, as compared to that for resource-rich languages, 0.20%. To further visualize the regularization effects, we present the learning curves for three representative languages, English (WSJ), French (UD-fr) and Romanian (UD-ro, low-resource), based on the development loss (see Figure 2). For all the three languages, we can observe that the AT model (red solid line) prevents overfitting better than the baseline (black dotted line), and this advantage is more significant in low resource languages. For example, in Romanian, the baseline model starts to increase development loss after 1,000 iterations even with dropout, whereas the AT model keeps improving until 2,500 iterations, achieving notably lower development loss (0.4 down). These results illustrate that AT can prevent overfitting especially well on small datasets and can augment the regularization power beyond dropout. AT can also be viewed as an effective means of data augmentation, where we generate and train with new examples the current model is particularly vulnerable to at every time step, enhancing the robustness of the model. AT can therefore be a promising tool to process low resource languages.

5 Analysis

In the previous sections, we demonstrated the regularization power of adversarial training (AT) on different languages, based on the overall POS tagging performance and learning curves. In this section, we conduct further analyses on the robustness of AT from NLP specific aspects such as word statistics, sequence modeling, downstream tasks, and word representation learning.

We find that AT can boost tagging accuracy on rare words and neighbors of unseen words (§5.1). Furthermore, this robustness against rare / unseen words leads to better sentence-level accuracy and downstream dependency parsing (§5.2). We illustrate these findings using two major languages, English (WSJ) and French (UD), which have substantially large training and testing data to discuss vocabulary statistics and sentence-level performance. Finally, we study the effects of AT on word representation learning (§5.3), and the applicability of AT to different sequential tasks (§5.4).

5.1 Word-level Analysis

Poor tagging accuracy on rare/unseen words is one of the bottlenecks in current POS taggers (Manning, 2011; Plank et al., 2016). Aiming to reveal

English (WSJ)

	Sentence-level Acc.	Stanford Parser UAS	Stanford Parser LAS	Parsey McParseface UAS	Parsey McParseface LAS
Baseline	59.08	91.53	89.30	91.68	87.92
Adversarial	**59.61**	**91.57**	**89.35**	**91.73**	**87.97**
(w/ gold tags)	–	(92.07)	(90.63)	(91.98)	(88.60)

French (UD)

	Sentence-level Acc.	Parsey Universal UAS	Parsey Universal LAS
Baseline	52.35	84.85	80.36
Adversarial	**53.36**	**85.01**	**80.55**
(w/ gold tags)	–	(85.05)	(80.75)

Table 5: Sentence-level accuracy and downstream dependency parsing performance by our baseline/adversarial POS taggers.

the effects of AT on rare/unseen words, we analyze tagging performance at the word level, considering vocabulary statistics.

Word frequency. To define rare/unseen words, we consider each word's frequency of occurrence in the training set. We categorize all words in the test set based on this frequency and study the test tagging accuracy for each group (see Table 3).[3] In both languages, the AT model achieves large improvements over the baseline on rare words (e.g., frequency 1-10 in training), as opposed to more frequent words. This result again corroborates the data augmentation power of AT under small training examples. On the other hand, we did not observe meaningful improvements on unseen words (frequency 0 in training). A possible explanation is that AT can facilitate the learning of words with at least a few occurrences in training (rare words), but is not particularly effective in inferring the POS tags of words for which no training examples are given (unseen words).

Neighboring words. One important characteristic of natural language tasks is the sequential nature of inputs (i.e., sequence of words), where each word influences the function of its neighboring words. Since our model uses BiLSTM-CRF for that reason, we also study the tagging performance on the neighbors of rare/unseen words, and analyze the effects of AT with the sequence model in mind. In Table 4, we cluster all words in the test set based on their frequency in training again, and consider the tagging accuracy on the neighbors (left and right) of these words in the test text. We observe that AT tends to achieve large improvements over the baseline on the neighbors of unseen words (training frequency 0), while the improvements on the neighbors of more frequent words remain moderate. Our AT model thus exhibits strong stability to uncertain neighbors, as compared to the baseline. We suspect that because we generate adversarial examples against entire input sentences, training with adversarial examples makes the model more robust not only to perturbations in each word but also to perturbations in its neighboring words, leading to greater stability to uncertain neighbors.

[3]To conduct the analysis, we picked the median result from the three repeated experiments.

5.2 Sentence-level & Downstream Analysis

In the word-level analysis, we showed that AT can boost tagging accuracy on rare words and the neighbors of unseen words, enhancing overall robustness on rare/unseen words. In this section, we discuss the benefit of our improved POS tagger in a major downstream task, dependency parsing.

Most of the recent state-of-the-art dependency parsers take predicted POS tags as input (e.g. Chen and Manning (2014); Andor et al. (2016); Dozat and Manning (2017)). Dozat et al. (2017) empirically show that their dependency parser gains significant improvements by using POS tags predicted by a Bi-LSTM POS tagger, while POS tags predicted by the UDPipe tagger (Straka et al., 2016) do not contribute to parsing performance as much. This observation illustrates that POS tagging performance has a great influence on dependency parsing, motivating the hypothesis that the POS tagging improvements gained from our adversarial training help dependency parsing.

To test the hypothesis, we consider three settings in dependency parsing of English and French: using POS tags predicted by the baseline model, using POS tags predicted by the AT model, and using gold POS tags. For English (PTB-WSJ), we first convert the treebank into Stanford Dependencies (SD) using Stanford CoreNLP (ver 3.8.0) (Manning et al., 2014), and then apply two well-known dependency parsers: Stanford Parser (ver 3.5.0) (Chen and Manning, 2014) and Parsey McParseface (SyntaxNet) (Andor et al., 2016). For French (UD), we use Parsey Universal from SyntaxNet. The three parsers are all publicly available and pre-trained on corresponding treebanks.

Table 5 shows the results of the experiments. We can observe improvements in both languages by using the POS tags predicted by our AT POS tagger. As Manning (2011) points out, when pre-

English (WSJ)

POS Cluster	NN	VB	JJ	RB	Avg.
1) Initial (GloVe)	0.243	0.426	0.220	0.549	0.359
2) Baseline	0.280	0.431	**0.309**	0.667	0.422
3) Adversarial	**0.281**	**0.436**	0.306	**0.675**	**0.424**

French (UD)

POS Cluster	NOUN	VERB	ADJ	ADV	Avg.
1) Initial (polyglot)	0.215	0.233	0.210	0.540	0.299
2) Baseline	0.258	0.271	0.262	0.701	0.373
3) Adversarial	**0.263**	**0.272**	**0.263**	**0.720**	**0.379**

Table 6: Cluster tightness evaluation for word embeddings, based on the cosine similarity measure. Higher scores indicate better clustering (cleaner word vector distribution). Each row corresponds to word vectors 1) at the beginning, 2) after baseline training, and 3) after adversarial training.

English (WSJ)

Perturbation scale α	0	0.001	0.01	0.05	0.1	0.5
Avg. cluster tightness	0.422	0.423	0.424	0.429	**0.436**	0.429

Table 7: Average cluster tightness for word embeddings trained with varied perturbation scale α (0 indicates baseline training).

dicted POS tags are used for downstream dependency parsing, a single bad mistake in a sentence can greatly damage the usefulness of the POS tagger. The robustness of our AT POS tagger against rare/unseen words helps to mitigate such an issue. This advantage can also be observed from the AT POS tagger's notably higher sentence-level accuracy than the baseline (see Table 5 left). Nonetheless, gold POS tags still yield better parsing results as compared to the baseline/AT POS taggers, supporting the claim that POS tagging needs further improvement for downstream tasks.

5.3 Effects on Representation Learning

Next, we perform an analysis on representation learning of words (word embeddings) for the English (PTB-WSJ) and French (UD) experiments. We hypothesize that adversarial training (AT) helps to learn better word embeddings so that the POS tag prediction of a word cannot be influenced by a small perturbation in the input embedding.

To verify this hypothesis, we cluster all words in the test set based on their correct POS tags[4] and evaluate the tightness of the word vector distribution within each cluster. We compare this clustering quality among the three settings: 1) beginning (initialized with GloVe or Polyglot), 2) after baseline training (50 epochs), and 3) after adversarial training (50 epochs), to study the effects of AT on word representation learning.

For evaluating the tightness of word vector distribution, we employ the cosine similarity metric, which is widely used as a measure of the closeness between two word vectors (e.g., Mikolov et al. (2013); Pennington et al. (2014)). To measure the tightness of each cluster, we compute the cosine similarity for every pair of words within, and then take the average. We also report the average tightness across all the clusters.

The evaluation results are summarized in Table 6. We report the tightness scores for the four major clusters: *noun*, *verb*, *adjective*, and *adverb* (from left to right). As can be seen from the table, for both languages, adversarial training (AT) results in cleaner word embedding distributions than the baseline, with a higher cosine similarity within each POS cluster, and with a clear advantage in the average tightness across all the clusters. In other words, the learned word vectors show stronger correlations with their POS tags. This result confirms that training with adversarial examples can help to learn cleaner word embeddings so that the meaning / grammatical function of a word cannot be altered by a small perturbation in its embedding. This analysis provides a means to interpret the robustness to input perturbations, from the perspective of NLP.

Relation with perturbation size ϵ. We also study how the size of added perturbations influences word representation learning in adversarial training. Recall that we set the norm of a perturbation ϵ to be $\alpha\sqrt{D}$, where D is the dimension of the concatenated input embeddings (see §3.2). For instance, $\alpha = 0$ would produce no noise; $\alpha = 1$ would generate a perturbation of a norm equivalent to the original word embeddings. We hypothesize that AT facilitates word representation learning when α is small enough to preserve the semantics of input words, but can hinder the learning when α is too large. To test the hypothesis, we repeat the clustering evaluation for word embeddings trained with varied perturbation scale α: 0, 0.001, 0.01, 0.05, 0.1, 0.5 (see Table 7). We observe that the quality of learned word embedding distribution keeps improving as α goes up from 0 to 0.1, but starts to drop around $\alpha = 0.5$. We also find that this optimal α in word embedding learning (i.e., 0.1) is larger than the α which

[4] We excluded words with multiple tags in the test text.

Model	F1
Tsuruoka et al. (2011)	93.81
Collobert et al. (2011)	94.32
Yang et al. (2017)	94.66
Suzuki and Isozaki (2008)	95.15
Søgaard and Goldberg (2016)	95.56
Hashimoto et al. (2017)	95.77
Peters et al. (2017)	**96.37**
Ours – Baseline (BiLSTM-CRF)	95.18
Ours – Adversarial	95.25

Table 8: Chunking F1 scores on the CoNLL-2000 task, with other top performing models.

Model	F1
Collobert et al. (2011)	89.59
Huang et al. (2015)	90.10
Chiu and Nichols (2016)	90.91
Lample et al. (2016)	90.94
Luo et al. (2015)	91.20
Ma and Hovy (2016)	91.21
Peters et al. (2017)	**91.93**
Ours – Baseline (BiLSTM-CRF)	91.22
Ours – Adversarial	91.56

Table 9: NER F1 scores on the CoNLL-2003 (English) task, with other top performing models.

yielded the best tagging performance on development sets (i.e., 0.01 or 0.05). A possible explanation is that while word embeddings can adapt to relatively large α (e.g., 0.1) during training, as adversarial perturbations are generated at the embedding level, such α could change the semantics of the input from the current tagging model's perspective and hinder the training of *tagging*.

5.4 Other Sequence Labeling Tasks

Finally, to further confirm the applicability of AT, we experiment with our BiLSTM-CRF AT model in different sequence labeling tasks: chunking and named entity recognition (NER).

Chunking can be performed as a sequence labeling task that assigns a chunking tag (B-NP, I-VP, etc.) to each word. We conduct experiments on the CoNLL 2000 shared task with the standard data split: PTB-WSJ Sections 15-18 for training and 20 for testing. We use Section 19 as the development set and employ the IOBES tagging scheme, following Hashimoto et al. (2017).

NER aims to assign an entity type to each word, such as *person*, *location*, *organization*, and *misc*. We conduct experiments on the CoNLL-2003 (English) shared task (Tjong Kim Sang and De Meulder, 2003), adopting the IOBES tagging scheme as in (Lample et al., 2016; Ma and Hovy, 2016).

The results are summarized in Table 8 and 9. AT enhanced F1 score from the baseline BiLSTM-CRF model's 95.18 to 95.25 for chunking, and from 91.22 to 91.56 for NER, also significantly outperforming Ma and Hovy (2016). These improvements made by AT are bigger than that for English POS tagging, most likely due to the larger room for improvement in chunking and NER. The improvements are again statistically significant, with p-value < 0.05 on the t-test. The experimental results suggest that the proposed adversarial training scheme is generally effective across different sequence labeling tasks.

Our BiLSTM-CRF AT model did not reach the performance by Hashimoto et al. (2017)'s multi-task model and Peters et al. (2017)'s state-of-the-art system that incorporates pretrained language models. It would be interesting future work to combine the strengths of these joint models (e.g., syntactic and semantic aids) and adversarial training (e.g., robustness).

6 Conclusion

We proposed and carefully analyzed a POS tagging model that exploits adversarial training (AT). In our multilingual experiments, we find that AT achieves substantial improvements on all the languages tested, especially on low resource ones. AT also enhances the robustness to rare/unseen words and sentence-level accuracy, alleviating the major issues of current POS taggers, and contributing to the downstream task, dependency parsing. Furthermore, our analyses on different languages, word/neighbor statistics and word representation learning reveal the effects of AT from the perspective of NLP. The proposed AT model is applicable to general sequence labeling tasks. This work therefore provides a strong basis and motivation for utilizing AT in natural language tasks.

Acknowledgements

We would like to thank Rui Zhang, Jonathan Kummerfeld, Yutaro Yamada, as well as all the anonymous reviewers for their helpful feedback and suggestions on this work.

References

Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations for multilingual nlp. In *CoNLL*.

Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav

Petrov, and Michael Collins. 2016. Globally normalized transition-based neural networks. In *ACL*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Gábor Berend. 2017. Sparse coding of neural word embeddings for multilingual sequence labeling. *TACL* .

Rich Caruana, Steve Lawrence, and C Lee Giles. 2001. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In *NIPS*.

Danqi Chen and Christopher D Manning. 2014. A fast and accurate dependency parser using neural networks. In *EMNLP*.

Jason PC Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. In *TACL*.

Michael Collins. 2002. Discriminative training methods for hidden markov models. In *EMNLP*.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research* 12:2493–2537.

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

Timothy Dozat, Peng Qi, and Christopher D. Manning. 2017. Stanford's graph-based neural dependency parser at the conll 2017 shared task. In *CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. pages 20–30.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *NIPS*.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *ICLR*.

Tao Gui, Qi Zhang, Haoran Huang, Minlong Peng, and Xuanjing Huang. 2017. Part-of-speech tagging for twitter with adversarial neural networks. In *EMNLP*.

Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. 2017. A joint many-task model: Growing a neural network for multiple NLP tasks. In *EMNLP*.

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

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991* .

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *ACL*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *EMNLP*.

Joo-Kyung Kim, Young-Bum Kim, Ruhi Sarikaya, and Eric Fosler-Lussier. 2017. Cross-lingual transfer learning for pos tagging without cross-lingual resources. In *EMNLP*.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *NAACL*.

Jiwei Li, Will Monroe, Tianlin Shi, Alan Ritter, and Dan Jurafsky. 2017. Adversarial learning for neural dialogue generation. In *EMNLP*.

Wang Ling, Tiago Luís, Luís Marujo, Rámon Fernandez Astudillo, Silvio Amir, Chris Dyer, Alan W Black, and Isabel Trancoso. 2015. Finding function in form: Compositional character models for open vocabulary word representation. In *EMNLP*.

Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. 2017. Delving into transferable adversarial examples and black-box attacks. In *ICLR*.

Gang Luo, Xiaojiang Huang, Chin-Yew Lin, and Zaiqing Nie. 2015. Joint entity recognition and disambiguation. In *EMNLP*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *ACL*.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*.

Christopher D Manning. 2011. Part-of-speech tagging from 97% to 100%: is it time for some linguistics? *Computational Linguistics and Intelligent Text Processing* pages 171–189.

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

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*.

Takeru Miyato, Andrew M Dai, and Ian Goodfellow. 2017. Adversarial training methods for semi-supervised text classification. In *ICLR*.

Dat Quoc Nguyen, Mark Dras, and Mark Johnson. 2017. A Novel Neural Network Model for Joint POS Tagging and Graph-based Dependency Parsing. In *CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*.

Joakim Nivre, Željko Agić, Maria Jesus Aranzabe, Masayuki Asahara, Aitziber Atutxa, Miguel Ballesteros, John Bauer, Kepa Bengoetxea, Riyaz Ahmad Bhat, Cristina Bosco, Sam Bowman, Giuseppe G. A. Celano, Miriam Connor, Marie-Catherine de Marneffe, Arantza Diaz de Ilarraza, Kaja Dobrovoljc, Timothy Dozat, Tomaž Erjavec, Richárd Farkas, Jennifer Foster, Daniel Galbraith, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Yoav Goldberg, Berta Gonzales, Bruno Guillaume, Jan Hajič, Dag Haug, Radu Ion, Elena Irimia, Anders Johannsen, Hiroshi Kanayama, Jenna Kanerva, Simon Krek, Veronika Laippala, Alessandro Lenci, Nikola Ljubešić, Teresa Lynn, Christopher Manning, Cătălina Mărănduc, David Mareček, Héctor Martínez Alonso, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Anna Missilä, Verginica Mititelu, Yusuke Miyao, Simonetta Montemagni, Shunsuke Mori, Hanna Nurmi, Petya Osenova, Lilja Øvrelid, Elena Pascual, Marco Passarotti, Cenel-Augusto Perez, Slav Petrov, Jussi Piitulainen, Barbara Plank, Martin Popel, Prokopis Prokopidis, Sampo Pyysalo, Loganathan Ramasamy, Rudolf Rosa, Shadi Saleh, Sebastian Schuster, Wolfgang Seeker, Mojgan Seraji, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Kiril Simov, Aaron Smith, Jan Štěpánek, Alane Suhr, Zsolt Szántó, Takaaki Tanaka, Reut Tsarfaty, Sumire Uematsu, Larraitz Uria, Viktor Varga, Veronika Vincze, Zdeněk Žabokrtský, Daniel Zeman, and Hanzhi Zhu. 2015. Universal dependencies 1.2. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2012. On the difficulty of training recurrent neural networks. *arXiv preprint arXiv:1211.5063* .

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Matthew E Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. In *ACL*.

Barbara Plank, Anders Søgaard, and Yoav Goldberg. 2016. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In *ACL*.

M. Schuster and K.K. Paliwal. 1997. Bidirectional recurrent neural networks. *Trans. Sig. Proc.* 45(11):2673–2681.

Uri Shaham, Yutaro Yamada, and Sahand Negahban. 2015. Understanding adversarial training: Increasing local stability of neural nets through robust optimization. *arXiv preprint arXiv:1511.05432* .

Anders Søgaard. 2011. Semi-supervised condensed nearest neighbor for part-of-speech tagging. In *ACL-HLT*.

Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *ACL*.

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.

Milan Straka, Jan Hajic, and Jana Straková. 2016. Udpipe: Trainable pipeline for processing conll-u files performing tokenization, morphological analysis, pos tagging and parsing. In *LREC*.

Jun Suzuki and Hideki Isozaki. 2008. Semi-supervised sequential labeling and segmentation using gigaword scale unlabeled data. *ACL-HLT* .

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *ICLR*.

Erik F Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *CoNLL*.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *HLT-NAACL*.

Yoshimasa Tsuruoka, Yusuke Miyao, and Jun'ichi Kazama. 2011. Learning with lookahead: Can history-based models rival globally optimized models? In *CoNLL*.

Beilun Wang, Ji Gao, and Yanjun Qi. 2017. A theoretical framework for robustness of (deep) classifiers against adversarial samples. In *ICLR*.

Yi Wu, David Bamman, and Stuart Russell. 2017. Adversarial training for relation extraction. In *EMNLP*.

Ziang Xie, Sida I Wang, Jiwei Li, Daniel Levy, Aiming Nie, Dan Jurafsky, and Andrew Y Ng. 2017. Data noising as smoothing in neural network language models. In *ICLR*.

Zhilin Yang, Ruslan Salakhutdinov, and William W. Cohen. 2017. Transfer learning for sequence tagging with hierarchical recurrent networks. In *ICLR*.