SeqVAT: Virtual Adversarial Training for Semi-Supervised Sequence Labeling

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

Abstract

Virtual adversarial training (VAT) is a powerful technique to improve model robustness in both supervised and semi-supervised settings. It is effective and can be easily adopted on lots of image classification and text classification tasks. However, its benefits to sequence labeling tasks such as named entity recognition (NER) have not been shown as significant, mostly, because the previous approach can not combine VAT with the conditional random field (CRF). CRF can significantly boost accuracy for sequence models by putting constraints on label transitions, which makes it an essential component in most state-of-the-art sequence labeling model architectures. In this paper, we propose SeqVAT, a method which naturally applies VAT to sequence labeling models with CRF. Empirical studies show that SeqVAT not only significantly improves the sequence labeling performance over baselines under supervised settings, but also outperforms state-of-the-art approaches under semi-supervised settings.

1 Introduction

While having achieved great success on various computer vision and natural language processing tasks, deep neural networks, even state-of-the-art models, are usually vulnerable to tiny input perturbations (Szegedy et al., 2014; Goodfellow et al., 2015). To improve the model robustness against perturbations, Goodfellow et al. (2015) proposed to train neural networks on both original training examples and adversarial examples (examples generated by adding small but worst-case perturbations to the original examples). This approach, named adversarial training (AT), has been reported to be highly effective on image classification (Goodfellow et al., 2015), text classification (Miyato et al., 2017), as well as sequence labeling (Yasunaga et al., 2018).

However, AT is limited to a supervised scenario, which uses the labels to compute adversarial losses. To make use of unlabeled data, virtual adversarial training (VAT) was proposed to extend AT to semi-supervised settings (Miyato et al., 2019). Unlike AT which treats adversarial examples as new training instances that have the same labels as original examples, VAT minimizes the KL divergence between estimated label distribution of original examples and that of adversarial examples. In this manner, both labeled and unlabeled data can be used in training to improve accuracy and robustness. As a semi-supervised learning algorithm, VAT was reported to be effective on both image (Goodfellow et al., 2015; Miyato et al., 2019) and text classifications (Miyato et al., 2017). Moreover, a recent study (Oliver et al., 2018) conducted comprehensive comparisons on various popular semi-supervised learning algorithms. VAT turned out to be the most effective one.

Despite its success in classification tasks, VAT has not shown similar effectiveness in sequence labeling tasks. In the conventional classification task, the model learns a mapping between a sentence (sequence of tokens) and a label. Nevertheless, in sequence labeling task, the target function becomes a mapping from a sequence of tokens to a sequence of labels. To apply VAT on sequence labeling, Clark et al. (2018) proposed to use a softmax layer on the top of token representations to obtain label probability distributions for each token. In this fashion, VAT could take KL divergence between tokens at the same position of the original sequence and the adversarial sequence as the adversarial losses. This approach shows marginal improvements over baseline models on several benchmarks, but fails to achieve comparable performance as other state-of-the-art models (Clark et al., 2018; Akbik et al., 2018; Peters et al., 2018; Devlin et al., 2019).

Although the approach above applies VAT on the entire sequence, it locally normalizes the label probability per token and assumes all transitions

between labels have equal possibilities. But in sequence labeling tasks, label transition probabilities are not always the same. For example, a song name is more likely to appear after a singer name, compared to a travel company.

To incorporate label transitions into sequence models, Lafferty et al. (2001) proposed conditional random field (CRF). CRF models the probability distribution of the whole label sequence given the input sequence, instead of yielding a label probability distribution for each token. It takes account of both token features and transition features. Most state-of-the-art sequence labeling models apply a CRF on top of token representations as a decoder. Such neural-CRF models usually outperform models without CRF (Ma and Hovy, 2016; Akbik et al., 2018; Peters et al., 2018; Yasunaga et al., 2018).

To apply the conventional VAT on a model with CRF, one can calculate the KL divergence on the label distribution of each token between the original examples and adversarial examples. However, it is sub-optimal because the transition probabilities are not taken into account.

To better address these issues, we proposed SeqVAT, a variant of VAT that can be used along with CRF. Our evaluation demonstrates that SeqVAT brings significant improvements in supervised settings, rather than marginal improvements reported from previous VAT-based approaches Clark et al.. In the semi-supervised settings, SeqVAT also outperforms many widely used methods such as self-training (ST) (Yarowsky, 1995) and entropy minimization (EM) (Grandvalet and Bengio, 2004), as well as the state-of-the-art semi-supervised sequence labeling algorithm, cross-view training (CVT) (Clark et al., 2018).

2 Related Work

2.1 Sequence Labeling

Sequence labeling is a series of common natural language processing tasks that predicts a label for each token within a sequence, rather than a label for the whole sequence. Such tasks include named entity recognition, chunking and part-of-speech (POS) tagging etc. Most state-of-the-art sequence labeling models are based on a neural-CRF architecture (Ma and Hovy, 2016; Akbik et al., 2018; Peters et al., 2018; Yasunaga et al., 2018). More precisely, the general design is to use bidirectional recurrent neural network (RNN) layers for encoding and a CRF layer for decoding. In addition, usually one or more convolutional neural network (CNN) or RNN layers are applied before the neural-CRF architecture to encode character-level information as part of the input. In this paper, we adapt the neural-CRF architecture by a CNN-LSTM-CRF model, which consists of one CNN layer to generate character embeddings, two layers of bidirectional long short-term memory (LSTM) as the encoder and a CRF layer as the decoder.

2.2 Semi-Supervised Learning

Semi-supervised learning is an important approach to improve model performance without enough labeled data. It utilizes unlabeled data to get more information which might be beneficial for supervised tasks. For semi-supervised learning, two robust and widely used approaches are self-training (ST) (Yarowsky, 1995) and entropy minimization (EM) (Grandvalet and Bengio, 2004). In natural language processing, ST has been successfully applied to word sense disambiguation (Yarowsky, 1995) and parsing (McClosky et al., 2006), and EM also has successful application in text classification (Sachan et al., 2019).

Recently, a powerful semi-supervised approach, cross-view training (CVT), has achieved state-of-the-art on several semi-supervised language tasks, including dependency parsing, machine translation and chunking (Clark et al., 2018). CVT forces the model to make consistent predictions when using the full input or partial input. Hence, it does not require label information and can be used for semi-supervised learning. In order to validate the effectiveness of our approach on semi-supervised sequence labeling, we make fair comparisons to those three semi-supervised learning methods in the experiments.

2.3 Virtual Adversarial Training

Adversarial training (Goodfellow et al., 2015) is a regularization method that enhances model robustness against input perturbations. It generates adversarial examples by injecting worst-case perturbations bounded by a small norm into the original examples, and adds them into training. As a consequence, model predictions would be consistent regardless of the perturbations. Prior to AT, several papers investigated various ways of perturbations (Xie et al., 2017). Adversarial training was demonstrated to be more effective since it introduces the perturbations which leading to the largest increase on model loss, respective to a constrained

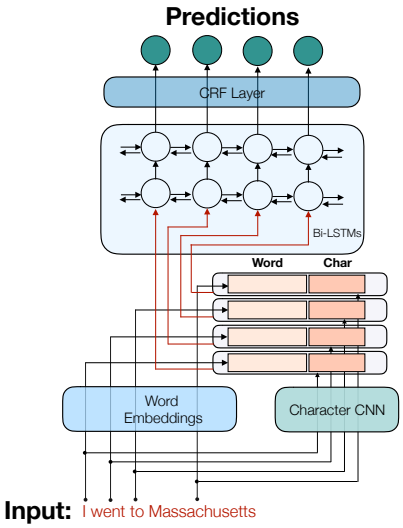


Figure 1: Sequence Labeling Model Architecture.

size (Goodfellow et al., 2015). Goodfellow et al. (2015) proved the effect of adversarial training in enhancing model robustness especially towards unseen samples for image classification. In addition to computer vision tasks, adversarial training also demonstrated its effectiveness on language tasks, such as text classification, POS tagging, named entity recognition and chunking (Miyato et al., 2017; Yasunaga et al., 2018).

To extend AT to semi-supervised settings, Miyato et al. (2019) proposed virtual adversarial training (VAT). "Virtual" means label information is not required in this new adversarial training approach and consequently it could be applied to both labeled or unlabeled training instances. VAT achieved state-of-the-art performance for image classification tasks (Miyato et al., 2019), and proved to be more efficient than traditional semi-supervised approaches, such as entropy minimization (Grandvalet and Bengio, 2004) and self-training (Yarowsky, 1995), from a recent study (Oliver et al., 2018).

However, despite the successful applications on text classification (Miyato et al., 2017), VAT has not shown great benefits to semi-supervised sequence labeling tasks, due to its incompatibility with CRF. In this paper, SeqVAT is proposed to make VAT compatible with CRF, and achieves significant improvements in sequence labeling.

3 Method

3.1 Model Architecture

Our baseline model architecture is illustrated in Fig.1. It adopts the basic architecture for several state-of-the-art sequence labeling models (Ma and Hovy, 2016; Peters et al., 2017; Akbik et al., 2018; Peters et al., 2018), called CNN-LSTM-CRF (CLC) in this paper. We apply a CNN layer to extract character information and concatenate its output with word embeddings as input features. Then, we feed the input features into LSTM layers, and decode with a CRF layer.

3.1.1 Word Embeddings

300-dimension randomly initialized word embeddings serve as word-level input. However, the model could learn embeddings with large norm, which makes the effects of adversarial perturbations with small norm insignificant (Miyato et al., 2017). To avoid such effect, we normalize the word embeddings at the beginning of each epoch. Denote $v = \{v_i | i = 1, 2, ..., n\}$ as the embeddings set, where n is vocabulary size, a specific embedding v_i is normalized by:

$$\hat{v_i} = \frac{v_i - E(v)}{\sqrt{D(v)}} \quad (1)$$

$$where \quad E(v) = \frac{1}{n}\sum_{i=1}^{n} v_i$$

$$and \quad D(v) = \frac{1}{n}\sum_{i=1}^{n}(v_i - E(v))^2$$

After normalization, word embeddings have zero mean and unit variance.

3.1.2 Character CNN Layer

Character-level information has proved to help improve the sequence labeling accuracy by capturing morphological features (Ma and Hovy, 2016). In this paper, 32-dimension embeddings are randomly initialized for each character. To ensure that adversarial perturbations have significant effects, character embeddings are also normalized at the beginning of each epoch in the same way as word embeddings. Suppose $u = \{u_i | i = 1, 2, ..., m\}$ where m is the number of unique characters show up in the dataset, a specific embedding u_i is randomly initialized and normalized by:

$$\hat{u_i} = \frac{u_i - E(u)}{\sqrt{D(u)}} \quad (2)$$

$$where \quad E(u) = \frac{1}{m}\sum_{i=1}^{m} u_i$$

$$and \quad D(u) = \frac{1}{m}\sum_{i=1}^{m}(u_i - E(u))^2$$

A CNN layer with 16 unigram, 16 bigram and 32 trigram filters is applied on top of all 32-dimension embeddings for one word. Hence, each word has 64-dimension character embeddings which are the output of CNN layer.

3.1.3 LSTM Layer

After concatenating character embeddings and word embeddings as input, all those features pass through two bidirectional LSTM layers with 256 neurons per direction to encode information for the whole sequence.

3.1.4 CRF Layer

To incorporate the probabilities of label transitions, the outputs of LSTM layers are fed into a linear-chain CRF decoder (Lafferty et al., 2001). Negative log-likelihood is computed as the training loss and Viterbi algorithm (Viterbi, 1967) is used for decoding.

3.2 Adversarial Training

Adversarial training (Goodfellow et al., 2015) is an effective method to improve model robustness over input perturbations. AT first generates adversarial examples, which are close to the original examples but model is not likely to correctly predict their labels (i.e. leading to most significant loss increase). Then, the model is trained with both original examples and adversarial examples. The loss on adversarial examples are treated as adversarial loss. In this paper, adversarial perturbations are added to word and character embeddings respectively. To prevent vanishing effects of adversarial perturbations explained in section 3.1.1 and 3.1.2, embeddings are normalized at the beginning of each epoch. Denote w and c as normalized word and character embeddings of the whole input sequence, θ is parameter of model, y is a vector of labels for all tokens in the sequence, and $Loss$ is the loss (i.e. negative log-likelihood) for the whole sequence. Given the bounded norms δ_w and δ_c respectively, the worst-case perturbations d_w and d_c for w and c are:

$$d_w = \underset{\epsilon,||\epsilon||_2 \leq \delta_w}{\operatorname{argmax}} Loss(y; w+\epsilon, c, \hat{\theta}) \quad (3)$$

$$d_c = \underset{\tau,||\tau||_2 \leq \delta_c}{\operatorname{argmax}} Loss(y; w, c+\tau, \hat{\theta}) \quad (4)$$

Note that all variables, y, w, c, d_w and d_c here are vectors for the whole sequence, since the last layer, CRF, is modeling the whole label sequence. In addition, $\hat{\theta}$ is current estimation of θ. The purpose for using constant value $\hat{\theta}$ instead of θ is to emphasize that the gradient should not propagate during generation of adversarial examples.

Hence, the worst-case perturbations d_w and d_c against current model can be calculated through (3) and (4) at each training step, and model can be trained on examples plus those perturbations to improve robustness against them. Yet, computing 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:

$$d_w = \frac{g_w}{||g_w||_2}\delta_w \quad (5)$$

$$d_c = \frac{g_c}{||g_c||_2}\delta_c \quad (6)$$

$$where \quad g_w = \nabla_w Loss(y; w, c, \hat{\theta}),$$

$$and \quad g_c = \nabla_c Loss(y; w, c, \hat{\theta})$$

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

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

3.3 Virtual Adversarial Training

Nevertheless, adversarial training cannot be applied to unlabeled data since label information is required to generate adversarial examples and compute adversarial loss. Virtual adversarial training is proposed (Miyato et al., 2019) to adapt adversarial training to semi-supervised settings. In VAT, instead of using the regular loss on perturbed examples as adversarial loss, the discrepancy (KL divergence) between predictions of original examples and those of adversarial examples acts as the adversarial loss. With this modification, label information is not needed in the computation of adversarial loss.

Indeed, the adversarial loss for VAT is written as:

$$
\begin{aligned}
L_{adv} &= KL(\ P_{ori}\ ||\ P_{adv}\ ) \qquad (8)\\
where\ P_{ori} &= P(\hat{y}; w, c, \hat{\theta}),\\
and\quad P_{adv} &= P(\hat{y}; w + d_w, c + d_c, \hat{\theta})
\end{aligned}
$$

Here, $\hat{y}$ is to emphasize that the computation of KL divergence takes current estimation of distribution over y, so that label information is not required. P_{ori} and P_{adv} are the estimated probability distributions of labels on original examples and adversarial examples respectively. As explained in section 1, VAT is not compatible with CRF. Hence, P_{ori} and P_{adv} here stand for sets of label distributions for tokens, computed by applying a softmax on top of LSTM output representations. As a consequence, the function P to estimate probability distributions of labels here is:

$$
P(\hat{y}; w, c, \hat{\theta}) = CLS(w, c, \hat{\theta}) \qquad (9)
$$

where CLS means applying softmax on top of CNN-LSTM encoder.

However, to compute worst-case perturbations d_w and d_c, label information y is still needed, as in equation (3), (4), (5) and (6). To get rid of the label information, the worst-case perturbations are now computed based on KL divergence between P_{ori} and P_{adv}, given the bounded norms δ_w and δ_c.

So word perturbation d_w is now defined by:

$$
\underset{\epsilon, ||\epsilon||_2 \leq \delta_w}{\text{argmax}}\ KL(P(\hat{y}; w, c, \hat{\theta})||P(\hat{y}; w + \epsilon, c, \hat{\theta})) \qquad (10)
$$

While character perturbation d_c is:

$$
\underset{\tau, ||\tau||_2 \leq \delta_c}{\text{argmax}}\ KL(P(\hat{y}; w, c, \hat{\theta})||P(\hat{y}; w, c + \tau, \hat{\theta})) \qquad (11)
$$

Those two computations are still intractable for gradient descent. By applying second-order approximation and a single iteration of power method, as in (Miyato et al., 2019), the word perturbation and character perturbation can be estimated with:

$$
\begin{aligned}
d_w &= \frac{g_w}{||g_w||_2}\delta_w \qquad (12)\\
d_c &= \frac{g_c}{||g_c||_2}\delta_c \qquad (13)
\end{aligned}
$$

where

$$
\begin{aligned}
g_w &= \nabla_\epsilon KL(P(\hat{y}; w, c, \hat{\theta})||P(\hat{y}; w + \epsilon, c, \hat{\theta})),\\
g_c &= \nabla_\tau KL(P(\hat{y}; w, c, \hat{\theta})||P(\hat{y}; w, c + \tau, \hat{\theta}))
\end{aligned}
$$

3.4 SeqVAT

Because of its incompatibility with CRF, adapting VAT to sequence labeling is not yet successful (Clark et al., 2018). To fully release the power of VAT to sequence labeling models with CRF, we propose a CRF-friendly VAT, named SeqVAT.

CRF models the conditional probability of the whole label sequence given the whole input sequence. Consequently, instead of using the label distribution over individual token, we could use the probability distribution for the whole label sequence, to compute KL divergence. The probability distribution can be denoted by:

$$
P(\hat{y}; w, c, \hat{\theta}) = CLC(w, c, \hat{\theta}) \qquad (14)
$$

where $\hat{y}$ is the whole label sequence, and CLC indicates the full CLC model.

Nevertheless, given a sequence with t tokens and l possible labels for each token, the total number of possible label sequences is l^t. Considering the substantial number of possible label sequences, it is not possible to compute the full probability distribution over all possible label sequences. To make the computation of such distribution possible, we estimate the full distribution by only considering the probabilities of k most possible label sequences, with one additional dimension to represent all the rest label sequences. Thus, the estimation of the probability distribution is $(k + 1)$ dimensions and feasible to compute.

To get the most possible label sequences, we apply a k-best Viterbi decoding (Huang and Chiang, 2005) on the original sequence in each training step. Denote $S = (s_1, s_2, .., s_k)$ as the k-best label sequences of current input embeddings w and c, and p_{crf} as the function to get probability of a label sequence. Given the current parameters $\hat{\theta}$, the probability distribution estimation P' can be written as:

$$
\begin{aligned}
P'(S; w, c, \hat{\theta}) &= (p'_1, p'_2, .., p'_k, 1 - \sum_{i=1}^{k} p'_i) \qquad (15)\\
where\quad p'_i &= p_{crf}(s_i; w, c, \hat{\theta}), i \in [1, k]
\end{aligned}
$$

Then, P_{ori} and P_{adv} can be denoted as:

$$
\begin{aligned}
P_{ori} &= P'(S; w, c, \hat{\theta}) \qquad (16)\\
P_{adv} &= P'(S; w + d_w, c + d_c, \hat{\theta}) \qquad (17)
\end{aligned}
$$

Here, d_w and d_c can be computed using the same approximation as VAT by:

$$d_w = \frac{g_w}{||g_w||_2}\delta_w \quad (18)$$

$$d_c = \frac{g_c}{||g_c||_2}\delta_c \quad (19)$$

where:

$$g_w = \nabla_\epsilon KL(P'(S;w,c,\hat{\theta})||P'(S;w+\epsilon,c,\hat{\theta})),$$
$$g_c = \nabla_\tau KL(P'(S;w,c,\hat{\theta})||P'(S;w,c+\tau,\hat{\theta}))$$

The adversarial loss for SeqVAT can be computed by:

$$L_{adv} = KL(\ P_{ori}\ ||\ P_{adv}\ ) \quad (20)$$

3.5 Training with Adversarial Loss

Regardless of the adversarial training method we use (AT, VAT or SeqVAT), sequence labeling loss is computed for all labeled data at each training step:

$$L_{label} = Loss(y;w,c,\eta,\hat{\theta}) \quad (21)$$

In addition, in every training step, adversarial examples are generated and adversarial loss L_{adv} is calculated based on the corresponding adversarial training algorithm. To combine the sequence labeling loss and adversarial loss, the total loss is a summation of those two loss:

$$L_{total} = L_{label} + \lambda L_{adv} \quad (22)$$

Here, weight λ is introduced to balance the model accuracy (sequence labeling loss) and robustness (adversarial loss). This objective function is optimized with respect to θ.

Note, unlabeled data might be leveraged in VAT and SeqVAT, and they do not have sequence labeling loss due to lack of annotation. Hence, the sequence labeling loss L_{label} would be set to 0 for unlabeled data.

4 Experiment

4.1 Dataset

Our proposed method is evaluated on three datasets: CoNLL 2000 (Sang and Buchholz, 2000) for chunking, CoNLL 2003 (Sang and Meulder, 2003) for named entity recognition (NER) and an internal natural language understanding (NLU) dataset for slot filling.

For chunking and NER, One Billion Word Language Model Benchmark (Chelba et al., 2014) is used as unlabeled data pool for semi-supervised learning. Considering the relatively small size of those two datasets, we randomly sampled 1% of the benchmark as the unlabeled dataset. We still have 20 times more data than training sets of CoNLL 2000 and 2003. For slot filling, our NLU dataset contains labeled and unlabeled sentences for 6 domains (detailed information is shown in Table.1). We directly use the unlabeled data for semi-supervised experiments.

Domain	Labels	Train	Test	Unlabeled
Cook	66	306155	55368	416348
Joke	20	230835	10311	586509
Booking	32	121067	5691	218864
News	12	116841	9607	339790
Assist	15	164364	5922	199383
Sporting	14	26763	3119	16034

Table 1: Number of sentences and labels in our internal NLU dataset.

4.2 Experiment Settings

All parameters are randomly initialized. All hyperparameters are chosen by grid search on the development set. Variational dropout (Blum et al., 2015) with rate 0.2 is applied to the input and output of each LSTM layer. The perturbation sizes for word and character embeddings, δ_w and δ_c, are 0.4 and 0.2 respectively. The weight for adversarial loss (i.e. λ) is set to 0.6. k is set to 3 for CoNLL datasets and 9 for our NLU dataset.

Sequence labeling 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

All sequence labeling tasks are evaluated with "slot-F1" metric, which is used in CoNLL 2000 and CoNLL 2003 shared tasks (Sang and Buchholz, 2000; Sang and Meulder, 2003).

5.1 Supervised Sequence Labeling

We evaluate our proposed SeqVAT technique in supervised settings and compare the results with other techniques designed to improve model robustness, including AT (Miyato et al., 2017), VAT (Miyato et al., 2019) and CVT (Clark et al., 2018).

To demonstrate the effectiveness of CRF, we compare results from models with or without CRF using each training technique mentioned above.

Method	Cook	Joke	Booking	News	Assist	Sporting	CoNLL 2000	CoNLL 2003
Baseline w/o. CRF	88.25	87.39	92.32	89.55	83.55	82.73	94.68	90.59
AT w/o. CRF	**88.51**	**88.05**	**93.03**	89.36	**84.14**	**84.25**	**95.03**	90.95
VAT w/o. CRF	88.40	88.04	92.84	**89.75**	83.92	84.21	94.89	90.87
CVT w/o. CRF	88.47	87.99	92.97	89.41	83.50	84.07	94.76	**91.02**
Baseline	88.53	87.97	93.04	90.32	84.99	86.67	95.18	91.20
AT	**88.93**	88.32	93.21	90.46	85.26	87.66	95.30	91.63
VAT	88.62	88.19	93.11	90.38	85.05	87.20	95.21	91.55
CVT	88.86	88.24	93.18	90.36	85.12	87.63	95.26	91.47
SeqVAT	88.90	**88.46**	**93.23**	**90.81**	**85.28**	**87.79**	**95.45**	**91.76**
ST†	88.73	88.69	93.42	91.29	85.13	86.73	95.27	91.66
EM†	88.68	88.70	93.45	91.21	85.09	86.79	95.91	91.69
VAT†	88.92	88.30	93.66	91.34	84.97	87.82	96.12	91.70
CVT†	88.81	88.75	93.57	91.31	85.58	87.80	96.19	92.08
SeqVAT†	**89.05**	**88.87**	**93.74**	**91.57**	**85.86**	**88.43**	**96.34**	**92.27**

Table 2: Slot F1 on all domains and datasets. "w/o. CRF" indicates CRF is excluded in the model architecture. † indicates semi-supervised sequence labeling.

In Table.2, the first set of results corresponds to models without CRF, while the second utilizes CRF. Note, based on the characteristics of each training technique, the added adversarial loss varies. Since AT is compatible with CRF, and thus its adversarial loss is computed on top of CRF. But as explained in Sec.1, the adversarial loss of conventional VAT cannot be calculated on top of CRF. Consequently, VAT in the second set of Table.2 only applies CRF for label loss. It uses adversarial loss without CRF.

As shown in Table.2, regardless of the training techniques, models with CRF consistently perform better than those without it. This demonstrates that CRF is a crucial component in sequence labeling. Hence, we conduct the rest of our evaluation only on models with CRF.

Moreover, except that AT performs slightly better than SeqVAT in Cook domain, SeqVAT can outperform all approaches in all the other domains/datasets. All improvements of SeqVAT over other approaches are statistically significant (with p-value < 0.05 in t-test). Compared with VAT used by Clark et al. (2018), SeqVAT consistently shows more significant improvements, which indicates that SeqVAT is a better way of adopting virtual adversarial loss to sequence labeling.

5.2 Semi-Supervised Sequence Labeling

VAT has been proved to be very effective in semi-supervised learning (Oliver et al., 2018). Our proposed SeqVAT preserves the ability of utilizing unlabeled data. In this work, we also compare SeqVAT with two widely used semi-supervised learning algorithms: self-training (ST) (Yarowsky, 1995), entropy minimization (EM) (Grandvalet and Bengio, 2004), and one state-of-the-art semi-supervised sequence labeling approach, cross-view training (CVT) (Clark et al., 2018). Detailed results are tabulated in the third set of Table.2. From this comparison, SeqVAT consistently outperforms conventional VAT, ST, EM, and CVT. The improvements over other approaches are also statistically significant with p-value < 0.05. These results suggest that SeqVAT is also highly effective at utilizing unlabeled data.

5.3 K-best Selection in SeqVAT

To choose the optimal k in k-best decoding, we conduct experiments with different ks on supervised sequence labeling. The F1 score from each k is plotted in Fig.2. From these plots, we observe that each dataset has its own optimal k for SeqVAT, and there is no unique k that gives the best results across datasets.

To get a better generalization over all datasets and tasks, we avoid selecting the optimal k for each dataset/domain. However, different sources of language have different characteristics, including vocabulary, sentence length, syntax etc. Using the same k for different types of text might limit the effects of SeqVAT. To make a balance between generalization and effectiveness, we use different k for different types of text, but the same k for all datasets/domains with the same source. We use $k = 3$ for CoNLL 2000 and 2003 (news), and $k = 9$ for our internal NLU dataset (spoken language).

5.4 Impact of Unlabeled Data

To further understand the effect of unlabeled data in semi-supervised learning, we analyze the corre-

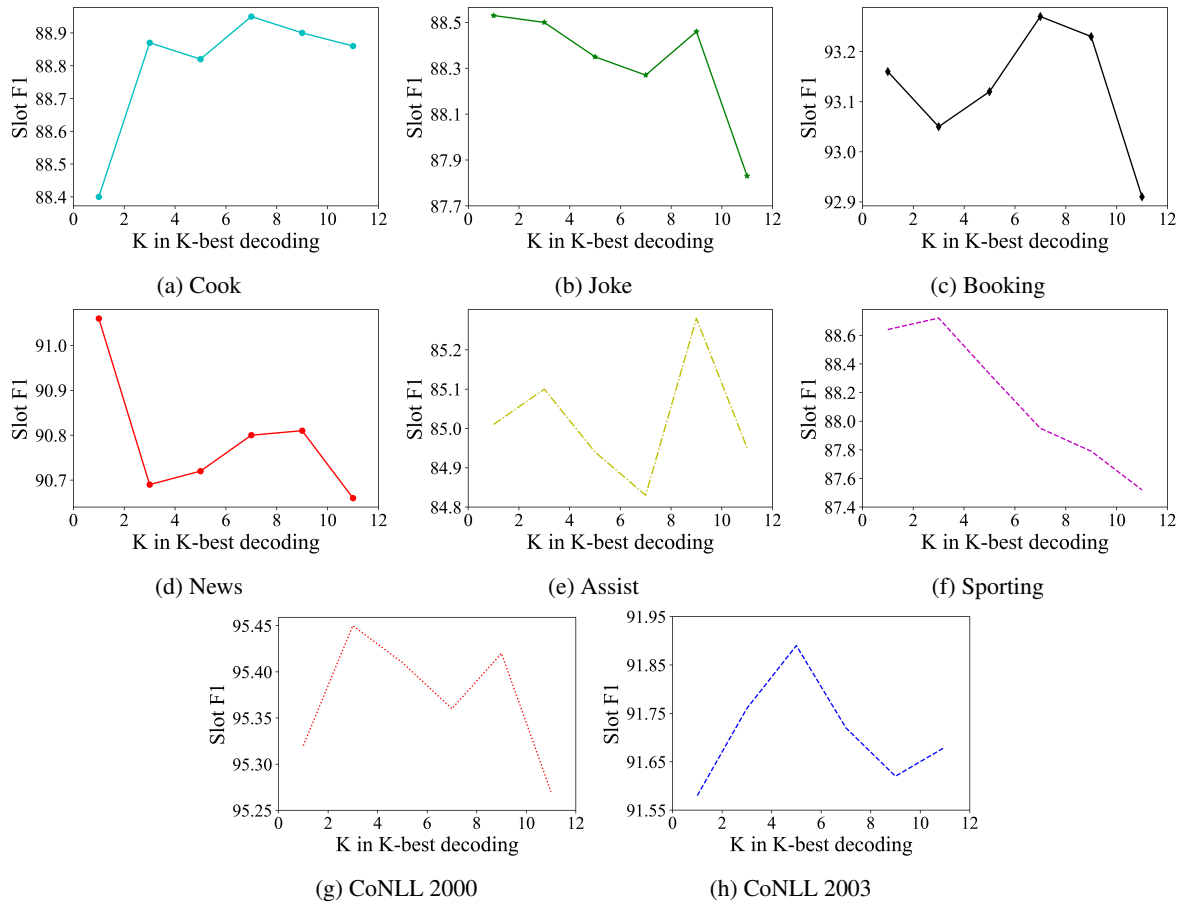


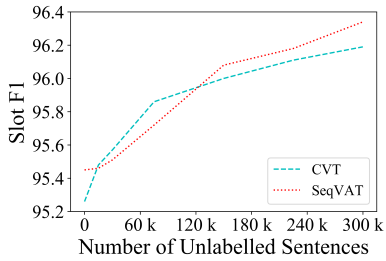
Figure 2: K-best results with different values of k for all domains and datasets.

lation between the amount of augmented unlabeled data and model performance on both CoNLL 2000 and 2003 datasets. For this analysis, we specifically focus ourselves on CVT and SeqVAT, which show the best accuracy across all datasets in Table.2. As shown in Fig.3, the amount of unlabeled data is a crucial factor for the performance of those two approaches. More specifically, the performance of those two approaches increases with more unlabeled data. For the CoNLL 2000 dataset, CVT has better performance when the unlabeled data is limited while SeqVAT gradually outperforms with more unlabeled data. As for the CoNLL 2003 dataset, SeqVAT shows consistently superior performance. This experiment shows that both approaches can provide significant benefits with a large amount of unlabeled data. In addition, SeqVAT has better utilization of unlabeled data, especially when having substantial unlabeled data.

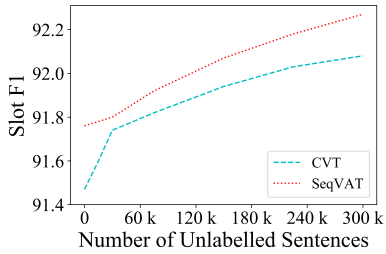
5.5 Comparison on Semi-Supervised Approaches

ST utilizes the unlabeled data by augmenting training data with the teacher model predictions, while EM makes the model more confident on the predictions for unlabeled data. Hence, both approaches are trying to force the model to trust predictions from the teacher model. If the teacher initially makes wrong predictions, the error would propagate to the student model.

Unlike them, CVT and VAT/SeqVAT construct similar sentences which might have the same labels, and force the model to make consistent predictions on them. If the model makes incorrect prediction for the original sentence, CVT and VAT/SeqVAT can form a "discussion" to reach an agreement among the prediction of the original sentence and that of the similar sentences. If the model can make correct predictions for some similar utterances, it would have a chance to fix the error. Consequently, CVT and VAT/SeqVAT are generally expected to be more effective than ST and EM on the use of unlabeled data.



(a) CoNLL 2000



(b) CoNLL 2003

Figure 3: Model performance with different amount of augmented unlabeled sentences.

The major difference between CVT and VAT is the mechanism of selecting similar sentences. CVT takes segments of the original sentence while VAT/SeqVAT generates new sentences by replacing tokens in the original sentence with their neighbors in the embedding space. Each approach has its own benefits and problems: 1) CVT can handle different tokens in the similar context, but would produce noise when the key words for meaning are not in the segments; 2) VAT generates truly similar sentences, but it might not be able to cover synonyms which have large distances in the embedding space. Hence, the effectiveness of them highly depends on the data. As in Table.2, CVT and VAT might outperform each other on different domains/datasets.

The improvements of SeqVAT over CVT and VAT can be explained by its compatibility with CRF, because CRF is a critical component for some sequence labeling tasks (including the three in this paper). The compatibility with CRF would largely affect the effectiveness of semi-supervised approaches. In other tasks where label transitions are important, we might not see significant gains from SeqVAT over VAT or CVT.

5.6 Insights from K-best Estimation

To make VAT compatible with CRF, we propose an idea to estimate the label sequence distribution using k-best estimation. This idea provides a view to optimize the label sequence level distribution directly rather than work on the label distribution per token. This idea could be beneficial for tasks needing distribution transfer on sequence models, such as knowledge distillation, multi-source transfer learning.

6 Conclusion

In this paper, we propose a CRF compatible VAT training algorithm and demonstrate that sequence labeling tasks can greatly benefit from it. Our proposed method, SeqVAT, has strong effects to improve model robustness and accuracy on supervised sequence labeling tasks. In addition, SeqVAT is also highly effective in semi-supervised settings and outperforms traditional semi-supervised algorithms (ST and EM) as well as a state-of-the-art approach (CVT). Overall, our approach is highly effective for chunking, NER and slot filling, and can be easily extended to solve other sequence labeling problems in both supervised and semi-supervised settings.

Acknowledgments

We want to thank all the anonymous reviewers for the helpful feedback and suggestions on this work.

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 1638–1649.

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.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2014. One billion word benchmark for measuring progress in statistical language modeling. In *INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association, Singapore, September 14-18, 2014*, pages 2635–2639.

Kevin Clark, Minh-Thang Luong, Christopher D. Manning, and Quoc V. Le. 2018. Semi-supervised sequence modeling with cross-view training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Bel-*

gium, October 31 - November 4, 2018, pages 1914–1925.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Yves Grandvalet and Yoshua Bengio. 2004. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems 17 [Neural Information Processing Systems, NIPS 2004, December 13-18, 2004, Vancouver, British Columbia, Canada]*, pages 529–536.

Liang Huang and David Chiang. 2005. Better k-best parsing. In *Proceedings of the Ninth International Workshop on Parsing Technology*, Parsing '05, pages 53–64, Stroudsburg, PA, USA. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, pages 282–289.

Xuezhe Ma and Eduard H. Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 4-9, 2006, New York, New York, USA*.

Takeru Miyato, Andrew M. Dai, and Ian J. Goodfellow. 2017. Adversarial training methods for semi-supervised text classification. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. 2019. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(8):1979–1993.

Avital Oliver, Augustus Odena, Colin Raffel, Ekin Dogus Cubuk, and Ian J. Goodfellow. 2018. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pages 3239–3250.

Matthew E. Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1756–1765.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237.

Devendra Singh Sachan, Manzil Zaheer, and Ruslan Salakhutdinov. 2019. Revisiting LSTM networks for semi-supervised text classification via mixed objective function. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6940–6948.

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.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and

Rob Fergus. 2014. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Andrew J. Viterbi. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Information Theory*, 13(2):260–269.

Ziang Xie, Sida I. Wang, Jiwei Li, Daniel Lévy, Aiming Nie, Dan Jurafsky, and Andrew Y. Ng. 2017. Data noising as smoothing in neural network language models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd Annual Meeting of the Association for Computational Linguistics, 26-30 June 1995, MIT, Cambridge, Massachusetts, USA, Proceedings*, pages 189–196. Morgan Kaufmann Publishers / ACL.

Michihiro Yasunaga, Jungo Kasai, and Dragomir R. Radev. 2018. Robust multilingual part-of-speech tagging via adversarial training. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 976–986.