Adversarial Training on Word–Char Embedding

Abebaw Tadesse* **and Joseph B. Collins†**

Abstract

In this work we propose a robust adversarial training model on hybrid word–char embeddings as developed in (Rei, Crichton, and Pyysalo 2016) based on the recent works of (Miyato, Dai, and Goodfellow 2016). The proposed neural training model addresses the existing critical issues with word–only embeddings which includes: poor vector representation for rare words and no representation for unseen words and the lack of proper mechanism to incorporate morphene–level informations that are not shared with the whole dictionary which, subsequently, leads to poor quality embeddings and hence low quality examples/adversarial examples. We present description of the proposed adversarial training model/architecture and addresses the implementation aspects at the word–char level. Our preliminary result on sequence labeling task on the First Certificate in English (FCE-PUBLIC) dataset (Yannakoudakis, Briscoe, and Medlock 2011) shows an improvement in accuracy of adversarial (regularized) training on word-char embedding over the baseline word-char embedding as well as on individual word/char-only and concatenated embeddings, as expected. The preliminary results also show that perturbation at word-char level yields better accuracy as compared to individual word-only and char-only perturbations.

Introduction

In this article we investigate the impact of adversarial training (Miyato, Dai, and Goodfellow 2016) on hybrid word-char embeddings, as developed in (Rei, Crichton, and Pyysalo 2016), on performance of Long Short-Term Memory (LSTM) based neural training models (Hochreiter and Schmidhuber 1997). In (Szegedy et al. 2013) and (Goodfellow, Shlens, and Szegedy 2014), it was shown that current neural models, particularly those that are linear or semi–linear w.r.t. the input are vulnerable to adversarial examples which are typically generated by simple linear, but carefully tuned perturbations of the input dataset. Additionally, in (Goodfellow, Shlens, and Szegedy 2014), it was demonstrated that adversarial training improves model performance at least in image classification tasks. In (Miyato, Dai, and Goodfellow 2016), the authors used adversarial and virtual adversarial (semi-supervised) training to improve a text or RNN models. Though word vector embeddings, in general, yield high quality vector representation for frequently seen words, they tend to produce poor quality word vectors for less frequent words and no embedding at all for previously unseen words (out of vocabulary representation), and character–level information is not shared with the whole dictionary (Rei, Crichton, and Pyysalo 2016). As a result, most of the time the generated example either does not change because there is no neighbor near enough, or else the perturbed context is not adversarial enough. In this work, we attempt to address these issues through hybrid implantation of word–char embedding under the settings described in (Rei, Crichton, and Pyysalo 2016) to develop a neural learning scheme for the generation and exploitation of adversarial examples in Natural Language Processing (NLP) contexts. We implement the proposed adversarial training model on LSTM based hybrid word-char embedding on a sequence labeling task on the FCE–PUBLIC (First Certificate in English) dataset (Yannakoudakis, Briscoe, and Medlock 2011). In section 2, a brief description of our proposed adversarial training model based on word–char embeddings followed by preliminary experimental results and discussions in Section 3.

Adversarial Training on the Word–Char Embedding Architecture

Word embeddings, in general, yield high quality distributional vector representation for frequently seen words, with semantically and functionally similar words having similar representations. However, they tend to produce poor quality word vectors for less frequent words and no embedding at all for previously unseen words (Out of Vocabulary words). Furthermore, there is no mechanism to exploit character–level patterns and sentimental words that are commonly unseen words in sentimental datasets such as Twit-

*A. Tadesse is with the Mathematics Dept., Langston University, Langston, Oklahoma, USA. e-mail: abebaw@langston.edu.

†J. Collins is with the Information Technology Division, Naval Research Laboratory, Washington D.C., USA. e-mail: joseph.collins@nrl.navy.mil

 In: Joseph Collins, Prithviraj Dasgupta, Ranjeev Mittu (eds.): Proceedings of the AAAI Fall 2018 Symposium on Adversary-Aware Learning Techniques and Trends in Cybersecurity, Arlington, VA, USA, 18-19 October, 2018, published at http://ceur-ws.org

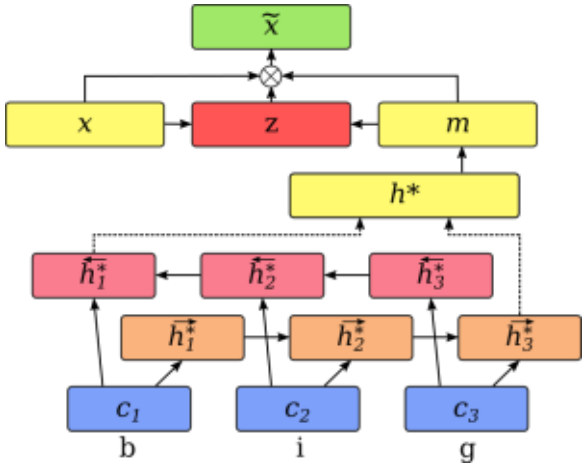

Figure 1: A bi–directional LSTM based hybrid word–char embedding (Extracted from (Rei, Crichton, and Pyysalo 2016))

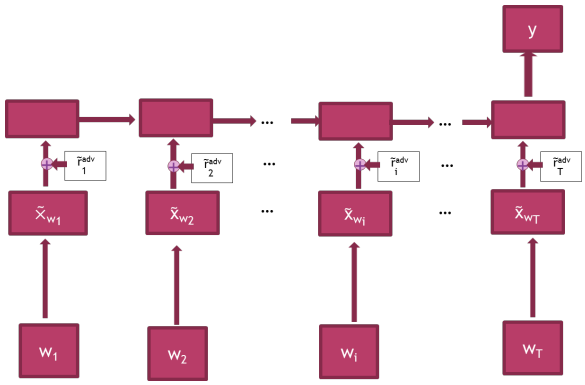

Figure 2: The proposed Architecture for Adversarial training on LSTM–based word–char embedding

ter datasets [1], and no immunity to typos (Rei, Crichton, and Pyysalo 2016). Consequently, the quality of adversarial examples generated using word-level only embeddings will inherit these weaknesses. In an attempt to address these critical issues we propose adversarial training on a bi–directional LSTM–based hybrid word-char architecture [Rei, Crichton, and Pyysalo2016)] as described in equation 1 below:

In the word–char embedding settings (Rei, Crichton, and Pyysalo 2016), a given word w will have dual vector representations, namely x_w and c_w as modeled in word2vec and the bidirectional char LSTM embeddings respectively. The hybrid architecture has a gating mechanism, also referred to as attention, which allows the model dynamically decide which level of information to tune into for each such word w in the dataset.

This will be achieved through two additional layers implementing the weight vector

$$z = \sigma(W_z^{(3)} \tanh(W_z^{(1)} x_w + W_z^{(2)} c_w)), \quad (1)$$

where $W_z^{(1)}, W_z^{(2)}, W_z^{(3)}$ respectively are weight matrices for calculating z, and σ is the sigmoid function. The hybrid embedding vector $\tilde{x}_w$ ($\tilde{x}$ in Figure 1) will then be expressed as the z–weighted sum of x_w (x in Figure 1) and c_w (m in Figure 1), given by

$$\tilde{x}_w = z * x_w + (1 - z) * c_w \quad (2)$$

(point–wise multiplication). The bidirectional LSTM realization of the character based word embedding m (Figure 1) is given by $m = \tanh(W_m h^*)$ where $h^* = [\overrightarrow{h_R^*}; \overleftarrow{h_L^*}]$ where the $\overrightarrow{h_R^*}$ and $\overleftarrow{h_L^*}$ are the extreme left and right hidden vectors (resp.) from each of the two LSTM components, namely $\overrightarrow{h_i^*} = LSTM(c_i, \overrightarrow{h_{i-1}^*})$ and $\overleftarrow{h_i^*} = LSTM(c_i, \overleftarrow{h_{i+1}^*})$, $i = 1, ...length(w)$. Furthermore, the attention–based architecture requires that the learned features in both word vectors $x_w$and c_w align. This will need to be incorporated as extra constraint on the loss function to encourage this agreement by optimizing

$$\tilde{\mathcal{J}} = \mathcal{J} + \sum_{k=1}^{T} g_k(1 - \cos(c_{w_k} x_{w_k})), \quad (3)$$

where $\mathcal{J}$ is the original embedding cost and $\tilde{\mathcal{J}}$ is the modified cost function and g_k is defined as $g_k(w_k) = 0$ for $w_k = OOV$ (Out Of Vocabulary words) and $g_k(w_k) = 1$ otherwise, $k = 1, .., T$ (T is the size of the input sequence (text)). Adversarial perturbation will then be applied on $\tilde{x_w}$, as implemented in (Rei, Crichton, and Pyysalo 2016) to generate its adversarial counterpart, $\tilde{x}_w^{adv}$, given by $\tilde{x}_w^{adv} = \tilde{x}_w + \tilde{r_w}^{adv}$ where $\tilde{r_w}^{adv} = \frac{\epsilon \nabla_{\tilde{x}_w} J(y|\tilde{x}_w, \theta)}{||\nabla_{\tilde{x}_w} J(y|\tilde{x}_w, \theta)||_2}$, $J(\tilde{x}_w, \theta)$ is the loss function (the negative loss likelihood function $-\log(p(y|x, \theta))$ for a classifier), θ is the parameter of the model (which should be viewed as a constant throughout the adversarial example generation process) and ϵ is the perturbation parameter. This needs to be done dynamically for each word vector $\tilde{x}_w$ to generate the needed adversarial examples. The aggregated adversarial perturbation on the concatenated sequence s (the labeled input text) of the (normalized) embedding vectors $[x^1, x^2, ..., x^T]$ is defined as $\tilde{r_s}^{adv} = \frac{\epsilon \nabla_s J(y|s,\theta)}{||\nabla_s J(y|s,\theta)||_2}$ and it's corresponding adversarial loss is defined as

$$J_{adv}(\theta) = -\frac{1}{N} \sum_{n=1}^{N} J(y_n, s_n + \tilde{r}_n^{adv}, \theta) \quad (4)$$

which will ensure robustness to the specified adversarial perturbation.

Here N denotes the number of labeled examples, $s_1, s_2, ...s_N$ are the input sequence of texts with corresponding labels $y_1, y_2, ...y_n$. For virtual adversarial training (semi-supervised training), following the formalism in (Miyato et al. 2015), we define the virtual adversarial perturbation as

$$\tilde{r}^{vadv} = \frac{\epsilon \nabla_{s+d} KL[p(., s, \theta)][p(., s + d, \theta)]}{||\nabla_{s+d} KL[p(., s, \theta)][p(., s + d, \theta)]||_2} \quad (5)$$

[1] Ashby, Charless, TensorFlow tutorial-analyzing Tweet's sentiment with character Level LSTM's, Deep Learning Blog, https://charlesashby.github.io/2017/06/05/sentiment-analyssi-withchar-lstm/

where $KL[p][q]$ denotes the KL divergence between distributions p and q.

The associated virtual adversarial loss will then be defined by

$$J_{vadv}(\theta) = \frac{1}{N'} \sum_{n=1}^{N'} KL[p(., s_n, \theta)][p(., s_n + \tilde{r}_n^{vadv}, \theta)] \tag{6}$$

where $\tilde{r_n}^{vadv}$ is the adversarial perturbation for the nth text (unlabeled) and N' is the number of such unlabeled texts (examples).

The crucial distinction between the adversarial (supervised) and the virtual adversarial (unsupervised) is that the perturbation (equation 5) and the loss function (equation 6) do not depend on the input labels which makes it applicable to unlabeled examples. (semi-supervised adversarial training). Furthermore, to regularize the flow of adversarial examples we use (Miyato, Dai, and Goodfellow 2016) the regularized adversarial loss

$$\tilde{J}(x, \theta) = \alpha J(x, \theta) + (1 - \alpha) J(x + \tilde{r}^{adv}, \theta) \tag{7}$$

(where $0 \leq \alpha \leq 1$ is the regularizing parameter) which will effectively make them resist and keep up with the current version of the model. The main of the paper is the proposal and preliminary testing of the adversarial training architecture on hybrid word–char embedding based on the existing framework (word–char embedding and adversarial training for semi–supervised text classification) as developed in (Rei, Crichton, and Pyysalo 2016) and (Miyato, Dai, and Goodfellow 2016). In the next section, we present the experimental settings and some preliminary results on a neural sequence labeling task on FCE–PUBLIC dataset (Yannakoudakis, Briscoe, and Medlock 2011).

Experiments on FCE–PUBLIC Dataset

The FCE–PUBLIC (for Error detection) dataset (Yannakoudakis, Briscoe, and Medlock 2011) (Rei and Yannakoudakis 2016) consists of 1141 examination Scripts for training, 97 examination Scripts for testing, 6 examination scripts for outliers experiments and 80 randomly selected scripts for developmental set. Tokens that have been annotated with an error tag are labeled as incorrect (i), otherwise, they are labeled as correct (c). The data is organized in a the Conference on Natural Language Learning (CoNLL) tab-separated format. Each line contains one token, followed by a tab and then the error label. With CoNLL format the dataset has 452833 train, 34599 developmental and 41477 test tokens. The total number of parameter count for the three representation are 2972052 (Word–based), 3452052 (Char concat) and 3152352 (Char attention) of which only a small fraction of the embeddings are utilized at every iteration. We performed the proposed adversarial trainings on Sequence Labeling (bidirectional LSTM) on word-char embedding on the FCE–PUBLIC Dataset [2]. The preliminary experimental results are briefly shown in Table 1 and Table 2. Table 1 presents performance of the regularized ($\alpha = 0.5$) adversarial training on word-char embedding on the dataset. The $F_{0.5}$–Score metric was used as an evaluation criterion as established in earlier works (Rei, Crichton, and Pyysalo 2016). The preliminary results (Table 1) shows an improvement in accuracy of adversarial (regularized) training on word-char embedding over both the baseline word-char embedding as well as on individual word/char-only and concatenated embeddings. Table 2 presents comparative accuracy results of the regularized adversarial training at the three representations levels (namely, word-only, char-only and word-char). These preliminary results show that perturbation at word-char level yields better accuracy as compared to individual word-only and char-only perturbation. Adversarial training at word-char level (Table 1 and Table 2) also performs better as compared to random perturbations as expected.

[2]We adopted here Tensorflow implementation of sequence labeling on FCE–PUBLIC dataset available at https://github.com/marekrei/sequence-labeler.

Table 1: Performance of Regularized Adversarial Training on Word-Char Embedding on FCE- PUBLIC Dataset. ($F_{0.5}$–Scores)

Word embedding:	Word-Only		Char-Only		Word-Char(Concact.)		Word-Char (attention)	
	Devt.	Test	Devt.	Test	Devt.	Test.	Devt.	Test
Baseline:	49.57	46.91	41.45	37.50	51.88	48.24	50.08	47.78
Random Perturbation:	52.24	48.49	52.99	49.63	53.01	50.01	52.92	49.74
Adv. Training(Regularized):	54.82	51.07	46.61	42.00	55.99	52.87	57.14	53.55

Table 2: Comparisons of regularized adversarial trainings at various perturbation levels (modes) on FCE–PUBLIC dataset. ($F_{0.5}$–Scores)

Pertubation Modes:	Word-Only Perturb		Char-Only Perturb		Word-Char(conc.) Perturb		Word-Char(attn.) perturb	
	Devt.	Test	Devt.	Test	Devt.	Test.	Devt.	Test
Random Perturbation:	53.70	50.33	53.07	49.74	53.01	50.01	52.92	49.74
Adv. Training (Regularized):	52.79	49.15	53.58	49.30	55.99	52.87	57.14	53.55

Conclusion

This work seeks to develop improved adversarial training model acting on word–char embeddings. It is well known that word–only/char–only embeddings have a major drawbacks in handling rare/unseen words and character–level information which subsequently leads to poor representation of valid and hence adversarial examples. The proposed adversarial training model is intended to overcome these challenges by applying the adversarial perturbation on word–char embeddings. It is envisioned that the proposed model along with adversarial regularization (i.e, fine tuning the parameter α) will bring significant improvements over the existing word–only/char–only adversarial training architectures. We performed some preliminary numerical experiments on the impact of regularized adversarial training on word–char embedding on a neural sequence labeling task on the FCE-PUBLIC dataset. Our preliminary result shows an improvement in accuracy of adversarial (regularized) training on word-char embedding over both the baseline word-char embedding as well as on individual word/char-only and concatenated embeddings. These preliminary results also show that perturbation at word-char level yields a better accuracy as compared to individual word-only and char-only perturbation. Further testing of the model need to be performed on several representative neural sequence labeling

and text classification tasks and various datasets.

Acknowledgment

We would like to thank Dr. Prithviraj Dasgupta, Dr. Ira S Moskowitz and Espiritu Hugo for their invaluable comments, suggestions and technical help during the progress of the research and the development of the paper.

References

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

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

Miyato, T.; Maeda, S.-i.; Koyama, M.; Nakae, K.; and Ishii, S. 2015. Distributional smoothing with virtual adversarial training. *arXiv preprint arXiv:1507.00677*.

Miyato, T.; Dai, A. M.; and Goodfellow, I. 2016. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*.

Rei, M., and Yannakoudakis, H. 2016. Compositional sequence labeling models for error detection in learner writing. *arXiv preprint arXiv:1607.06153*.

Rei, M.; Crichton, G. K.; and Pyysalo, S. 2016. Attending to characters in neural sequence labeling models. *arXiv preprint arXiv:1611.04361*.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Yannakoudakis, H.; Briscoe, T.; and Medlock, B. 2011. A new dataset and method for automatically grading esol texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, 180–189. Association for Computational Linguistics.