Unsupervised Paraphrasing Consistency Training for Low Resource Named Entity Recognition

Rui Wang
Duke University
rw161@duke.edu

Ricardo Henao
Duke University
ricardo.henao@duke.edu

Abstract

Unsupervised consistency training is a way of semi-supervised learning that encourages consistency in model predictions between the original and augmented data. For Named Entity Recognition (NER), existing approaches augment the input sequence with token replacement, assuming annotations on the replaced positions unchanged. In this paper, we explore the use of paraphrasing as a more principled data augmentation scheme for NER unsupervised consistency training. Specifically, we convert Conditional Random Field (CRF) into a multi-label classification module and encourage consistency on the entity appearance between the original and paraphrased sequences. Experiments show that our method is especially effective when annotations are limited.

1 Introduction

Supervised training for Named Entity Recognition (NER) requires token level annotations, which are time consuming and more expensive to obtain than the sequence level annotations commonly used for classification tasks. Due to the scarcity of labeled data, various semi-supervised approaches have been investigated for training in low-resource scenarios, *i.e.*, only a small amount of labeled data is available (Clark et al., 2018; Lowell et al., 2020).

Unsupervised consistency training is a common approach to semi-supervised learning in NER. It encourages prediction consistency between the original and the augmented examples, by leveraging the availability of a larger amount of unlabeled data. Recently, Xie et al. (2019) proposed the Unsupervised Data Augmentation (UDA), which substitutes traditional token-wise perturbations with higher quality data augmentation, *e.g.*, paraphrasing via back-translation. UDA achieves state-of-the-art results on a wide variety of classification tasks with only tens or hundreds of labeled examples, and even sometimes matching the performance of supervised training with a much larger (fully-annotated) dataset. In the case of NER, due to the difficulty of obtaining token-level annotations, it is of interest to extend UDA for NER models whose predictions are (token-level) sequences instead of single (sentence-level) labels.

More recently, Lowell et al. (2020) augmented unlabeled samples for NER by randomly replacing a portion of input tokens with outputs from a language model, thus constraining the model predictions to be invariant to the replacement operation. There are two problems with this approach: *i*) there is no guarantee that the type of entity (label) will remain unchanged after replacement; and *ii*) the newly generated context from replacement is constrained by length of the original sequence, which restricts the quality of augmentation. In fact, Xie et al. (2019) suggests that there exists strong correlation between the quality of the augmentation and the performance of consistency training.

In this paper, we explore the use of paraphrasing as a means for higher quality data augmentation for unsupervised consistency training in NER. Compared with token replacement, the key difficulty of using paraphrasing is that the alignment of tokens between the original and paraphrased sequence is unclear. However, since paraphrasing does not change the substance of the text, we can expect a paraphrase to contain the same entities as in the original sequence. So motivated, instead of relying on token-level consistency, we encourage consistency on the occurrence of entities between predictions on the original and paraphrased sequences. In doing so, we convert the Conditional Random Field (CRF) (Lafferty et al., 2001) predictor of NER into a binary multi-label classification module indicating the occurrence of each entity label, *e.g.*, location (LOC) or person (PER). Experimental results show that our method outperforms token replacement and other semi-supervised learning approaches when annotations are scarce.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 5303–5308
November 7–11, 2021. ©2021 Association for Computational Linguistics

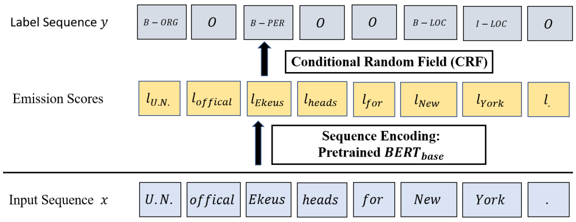


Figure 1: Illustration of our NER model.

2 Related Work

In addition to token replacement discussed above (Lowell et al., 2020), Şahin and Steedman (2019); Dai and Adel (2020) also investigated on randomly swapping tokens or text-spans in the input sequence as augmentation. However, such methods may be problematic for languages that rarely have inflectional morphemes, such as English, where words follow strict ordering (Şahin and Steedman, 2019). Therefore, we are not considering swap-based methods in our experiments. Other semi-supervised approaches for NER include CVT (Clark et al., 2018) which regularizes model predictions to be invariant when masking-out parts of the input data. Recently, Chen et al. (2020b) proposed an adapted version of virtual adversarial training with CRF, outperforming CVT on NER tasks. In the experiments, we show that our method can achieve better performance than both token replacement and SeqVAT in low-resource scenarios.

3 Methodology

3.1 The NER model

Following Beltagy et al. (2019), our model for NER consists of a $BERT_{base}$ () encoder and a CRF module for prediction. See Figure 1 as an illustration. Assume the input sequence $x = [x_1, \dots, x_T]$ and label sequence $y = [y_1, \dots, y_T]$, where T is the sequence length, and let N be the number of possible labels for any given token in NER. The output of $BERT_{base}$ is first projected into emission scores $l(x) = [l(x_1), \dots, l(x_T)]$, where each $l(x_t)$, for $t = 1, \dots, T$, is an N-dimensional vector containing scores for each class. Given $l(x)$, the CRF module generates the probability of predicting label sequence y, *i.e.*, $p_\theta(y|x)$, where θ denotes all model parameters.

Algorithm 1 Computing the normalization term in CRF.

```
Input: Assuming $T > 1$, $l \in R^{T \times N}$, $A \in R^{N \times N}$ and $s \in R^N$.
Output: $NM(l, A, s)$ as in eq (2).
Initialization: Let $l[t,:]$ be the $t$th row of $l$. Reshape $s \in R^{1 \times N}$. Initialize variable $p$ as $p = s + l[1,:]$.
for t=2,$\cdots$, T do
    $p = \log ColumnSum(\exp(p^T \mathbf{1}_N^T + A + \mathbf{1}_N l[t,:]))$
end for
$NM(l, A, s) = \log sum(\exp(p))$
```

CRF module: Let $[l(x_t)]_j$ be the j-th entry of the N-dimensional vector $l(x_t)$. Define $A \in R^{N \times N}$ as the transition matrix so A_{j_1, j_2} corresponds to the (unnormalized) score for the transition from label j_1 to label j_2, and $s \in R^N$ as the starting vector where its j-th element s_j is a score for $y_1 = j$. The prediction score for label sequence y accounting for transitions is given by

$$\begin{aligned} \text{score}(y, x) &= s_{y_1} + [l(x_1)]_{y_1} \\ &+ \sum_{t=2}^{T} \left(A_{y_{t-1}, y_t} + [l(x_t)]_{y_t} \right). \end{aligned} \tag{1}$$

The log likelihood of predicting y given x with the CRF can be evaluated by normalizing the scores in (1) by that of all possible label sequences, *i.e.*,

$$\begin{aligned} \log p_\theta(y|x) &= \text{score}(y, x) \\ &\underbrace{- \log \sum_{y' \in \mathcal{Y}} \exp\{\text{score}(y', x)\}}_{\log \mathbb{Q} = \text{NM}(l(x), A, s)}, \end{aligned} \tag{2}$$

where $\mathcal{Y}$ is the set of all possible label sequences of length T. The normalization factor in (2) can be computed with dynamic programming from $\{l(x), A, s\}$ as in Algorithm 1.

3.2 Unsupervised Consistency Training

Consider a small labeled NER dataset $D^l = \{X^l, Y^l\}$, where X^l and Y^l are collections of token sequences and label sequences, respectively. We seek to learn a NER model with D^l, by taking advantage of an external unlabeled dataset $D^u = \{X^u\}$. The learning objective for unsupervised consistency training is

$$\begin{aligned} L = &\lambda \mathbb{E}_{x^u \sim D^u}[L_c(x^u, q(x^u))] \\ &- \mathbb{E}_{(x^l, y^l) \sim D^l}[\log p_\theta(y^l|x^l)], \end{aligned} \tag{3}$$

where $L_c(\cdot, \cdot)$ is the consistency loss, λ is a balancing parameter and $q(\cdot)$ is the perturbation function used for augmentation. For instance, when $q(\cdot)$ is token replacement (Lowell et al., 2020), $L_c(\cdot,\cdot)$ penalizes the difference in predicted label distributions on each sequence position, *i.e.*,

$$L_c = \frac{1}{T}\sum_{t=1}^{T} \mathrm{KL}\left(p_\theta(y_t|x)||p_\theta(y_t|q(x))\right) \qquad (4)$$

where $\mathrm{KL}(\cdot||\cdot)$ is the Kullback-Leibler divergence.

3.3 Consistency with Paraphrasing

With the understanding of the concerns associated with token replacement discussed above, we propose using back-translation to paraphrase unlabeled sequences as an alternative way of data augmentation. Note that in (4), $q(\cdot)$ cannot be implemented as back-translation, provided the alignment of labels and tokens between x and $q(x)$, its paraphrased version, is unclear, *i.e.*, though all the entities in x should be also present in $q(x)$, their locations will likely be different, which makes using the token-wise consistency loss in (4) problematic. Therefore, we propose encouraging consistency on the prediction of entity *occurrences* between x and $q(x)$, rather than the consistency given the location as in (4). For instance, if a location (LOC) is predicted from x, one should expect to also see it predicted from $q(x)$. In doing so, we circumvent the token alignment issues between x and $q(x)$.

Specifically, we convert the CRF objective in (2) into a multi-label classification objective targeting the consistency of the occurrence of entity labels in augmented data. Assume we have a set G with M entity labels, *e.g.*, $G = \{LOC, PER, ORG, MISC\}$ in the CONLL2003 dataset. In the BIO format, each token can take labels in the set $\{O, B_e, I_e\}_{e\in G}$, where O represents an irrelevant token, B_e denotes the label for the beginning of entity e, and I_e is the label of a token belonging to entity e other than its first token, *e.g.*, B_{LOC} and I_{LOC} stands for $B-LOC$ and $I-LOC$. In this setting, token labels can take $N = 2M + 1$ distinct values from M entities of interest.

We can evaluate the likelihood of a sequence x containing entities of e as

$$\begin{aligned} p_e(x) &= \sum_{y\in\mathcal{S}_e} p_\theta(y|x) \\ &= 1 - \sum_{y\in\mathcal{Y}\backslash\mathcal{S}} \exp\{\mathrm{score}(y,x)\}/\mathbb{Q}, \qquad (5) \end{aligned}$$

where $\mathcal{S}_e = \{y|\exists t, y_t = I_e \vee y_t = B_e\}$, is the set of all label sequences containing at least one occurrence of entity e. Then, since $\mathrm{NM}(l(x), A, s)$ in (2) allows one to evaluate the likelihood of all possible label sequences given x, to evaluate (5) we can use (2) but by changing the sum over $\mathcal{Y}$ to be over $\mathcal{Y}\backslash\mathcal{S}$, so

$$p_e(x) = 1 - \exp\{\mathrm{NM}(l'(x), A', s')\}/\mathbb{Q}, \qquad (6)$$

where $l'(x) \in R^{T\times(N-2)}$, $A' \in R^{(N-2)\times(N-2)}$ and $s' \in R^{N-2}$ are entries of $l(x) \in R^{T\times N}$, $A \in R^{N\times N}$ and $s \in R^N$ without the dimensions corresponding to entity label e. Moreover, the consistency loss can be written as multi-label classification objective as

$$L_c = \sum_{e\in G} \mathrm{BCE}(p_e(x), p_e(q(x))), \qquad (7)$$

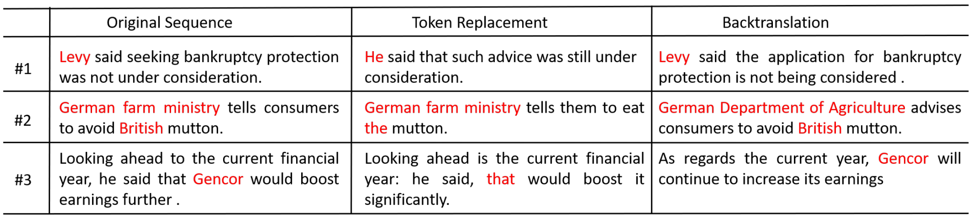
where $\mathrm{BCE}(\cdot,\cdot)$ is the binary cross-entropy loss. Note that i) (7) is a multi-label classification objective because it accounts for the fact that any given sequence x can have occurrences of multiple different entities; and ii) we have effectively adapted the CRF objective in (2) to a multi-label scenario where we encourage the consistence of the occurrence of the entities rather than the consistency of their locations as in (4).

4 Experiments

4.1 General Setup

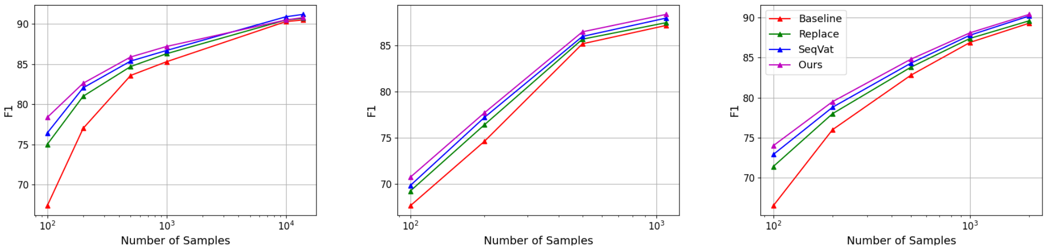
We focus the low resource scenario where there are only several hundred or one thousand labeled sequences. Following the implementation of Chen et al. (2020a), we use German as the pivot language for back-translation. For D^l, we choose three target datasets: CONLL2003, Wikigold and Wall Street Journal (WSJ). We use the full dataset of CoNLL2003 and Wikigold with entity LOC, PER, ORG and $MISC$. We randomly split Wikigold into 1096 for training, 200 for evaluation and 400 for testing. For low resource training, we only use a subset of 2K training instances from WSJ. Our unlabeled data is from the One Billion Word Benchmark (Chelba et al., 2013). To generate the unlabeled data, we train a binary text classifier based on the BERT model, distinguishing between the labeled dataset and the One Billion Word Benchmark. We use the publicly available pretrained De-En and En-De translation models from Huggingface[1] for back-translation. For all the

[1] https://github.com/huggingface/transformers



	Original Sequence	Token Replacement	Backtranslation
#1	Levy said seeking bankruptcy protection was not under consideration.	He said that such advice was still under consideration.	Levy said the application for bankruptcy protection is not being considered .
#2	German farm ministry tells consumers to avoid British mutton.	German farm ministry tells them to eat the mutton.	German Department of Agriculture advises consumers to avoid British mutton.
#3	Looking ahead to the current financial year, he said that Gencor would boost earnings further .	Looking ahead is the current financial year: he said, that would boost it significantly.	As regards the current year, Gencor will continue to increase its earnings

Figure 2: Augmented examples from CoNLL2003. Red denotes entities and their correspondence in augmentation.



(a) CoNLL2003 (14K)

(b) Wikigold (1.1K)

(c) WSJ (2K)

Figure 3: F1 scores with different amount of labeled data.

experiments, We training our BERT_CRF model with learning rate 5e-5 using Adam optimizer and linear learning rate scheduler. We set the balncing parameter $\lambda = 1$. Here, we introduce the definition of methods we are comapring with.

- *Baseline*: We train our model only using the labeled data, *i.e.*, without consistency training.
- *Token Replacement*: We implement the token replacement strategy as in Lowell et al. (2020), where the tokens are replace by outputs from language modeling with $BERT_{base}$.
- *SeqVat*: We compare with the recently proposed SeqVat (Chen et al., 2020b), which is a variant of Virtual Adversarial Training for the model with CRF.

4.2 Multi-label Classification *vs.* NER

Provided we do not count with sequence labels for the unlabeled data, we use the multi-label classification objective in (7) for consistency training as a substitute (proxy) for the NER objective in (2). One natural question is whether errors of the two objectives are related. Further, one may hypothesize that the performance of NER and the multi-label prediction of entity occurrences are not equally affected by sequence length. To examine this, we define *Error I* as test sequences for which the NER sequence labels are incorrectly predicted but multi-label predictions are correct, and denote *Error II* as the sequence for which, both NER and multi-label predictions are wrong.

With models trained on each target dataset (CONLL2003, Wikigold and WSJ), in Figure 4, we show the proportion of *Error II* relative to all errors (*Error I* and *II*) as a function of the test sequence length. We observe that: *i) Error II* accounts for the majority of the errors, *i.e.*, most errors in NER label sequences are also multi-label classification errors; and *ii)* the proportion of *Error II* decreases with sentence length, which is reasonable because predicting label sequences becomes more difficult as sequence length increases, whereas predicting entity label occurrences does not necessarily becomes more difficult. For instance, for long sentences that contain multiple occurrences of different types of entities, error in predicting one of the entities for NER may not affect the result of multi-label classification for appearance of entity labels. Motivated by the reasoning above and the results in Figure 4, we exclude sentences with length longer then 40 tokens when selecting D^u for consistency training.

4.3 Results

Figure 3 shows the results of consistency training with different amounts of labeled data. Every point is an average of three runs with different random seeds. We find that, though SeqVat may outperform the proposed model when there is a large amount of labeled data, our approach outperforms all competing approaches when the labeled data is scarce, *e.g.*

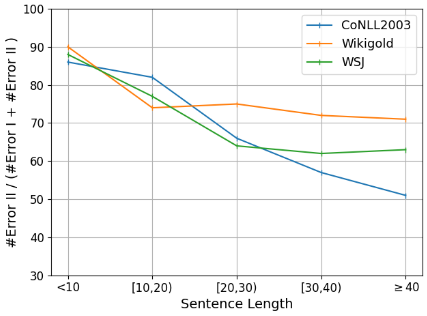


Figure 4: Percentage of *Error II* during testing.

several hundred or one thousand. Specifically, we have the F1 score improvement of 1.94 and 11.01 over SeqVat and Baseline, respectively, with 100 labeled instances for CoNLL2003. The improvements of the non-Baseline methods compared to Baseline is less obvious on Wikigold, probability because BERT model has been pretrained on the Wikipedia corpus. Note that in agreement with the experimental results in Xie et al. (2019) on classification tasks, our results with back-translation perform consistently better than token replacement, further supporting the value of high-quality augmentations for consistency training.

In Figure 2, we show examples of different augmentations. We find that the token labels can be changed after replacement, *e.g.* “Levy" ($B-PER$) to “He" (O). Also, there may sometimes be unnecessary punctuation in the generated context (#3). Alternatively, our paraphrasing with back-translation tend to keep the entities in the original sequence, generating new context that is not constrained by the original sequence length.

5 Conclusion

In this paper, we explored the use of paraphrasing as data augmentation strategy in unsupervised consistency training for NER. Experiments show that our approach outperforms token replacement and another state-of-the-art semi-supervised learning approach in low-resource scenarios.

References

M Saiful Bari, Shafiq Joty, and Prathyusha Jwalapuram. 2020. Zero-resource cross-lingual named entity recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7415–7423.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2013. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*.

Jiaao Chen, Zichao Yang, and Diyi Yang. 2020a. Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification. *arXiv preprint arXiv:2004.12239*.

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

Kevin Clark, Minh-Thang Luong, Christopher D Manning, and Quoc V Le. 2018. Semi-supervised sequence modeling with cross-view training. *arXiv preprint arXiv:1809.08370*.

Ryan Cotterell and Kevin Duh. 2017. Low-resource named entity recognition with cross-lingual, character-level neural conditional random fields. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 91–96.

Xiang Dai and Heike Adel. 2020. An analysis of simple data augmentation for named entity recognition. *arXiv preprint arXiv:2010.11683*.

Jan Vium Enghoff, Søren Harrison, and Željko Agić. 2018. Low-resource named entity recognition via multi-source projection: Not quite there yet? In *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, pages 195–201.

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

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

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

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

David Lowell, Brian E Howard, Zachary C Lipton, and Byron C Wallace. 2020. Unsupervised data augmentation with naive augmentation and without unlabeled data. *arXiv preprint arXiv:2010.11966*.

Gözde Gül Şahin and Mark Steedman. 2019. Data augmentation via dependency tree morphing for low-resource languages. *arXiv preprint arXiv:1903.09460*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. 2019. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*.