Local Additivity Based Data Augmentation for Semi-supervised NER

Jiaao Chen*, **Zhenghui Wang***, **Ran Tian**[1], **Zichao Yang**[2], **Diyi Yang**
Georgia Institute of Technology, [1]ASIT Japan / Google, [2]Citadel Securities
{jchen896,zhwang,dyang888}@gatech.edu

Abstract

Named Entity Recognition (NER) is one of the first stages in deep language understanding yet current NER models heavily rely on human-annotated data. In this work, to alleviate the dependence on labeled data, we propose a **L**ocal **A**dditivity based **D**ata **A**ugmentation (LADA) method for semi-supervised NER, in which we create virtual samples by interpolating sequences *close* to each other. Our approach has two variations: Intra-LADA and Inter-LADA, where Intra-LADA performs interpolations among tokens within one sentence, and Inter-LADA samples different sentences to interpolate. Through linear additions between sampled training data, LADA creates an infinite amount of labeled data and improves both entity and context learning. We further extend LADA to the semi-supervised setting by designing a novel consistency loss for unlabeled data. Experiments conducted on two NER benchmarks demonstrate the effectiveness of our methods over several strong baselines. We have publicly released our code at https://github.com/GT-SALT/LADA.

1 Introduction

Named Entity Recognition (NER) that aims to detect the semantic category of entities (e.g., persons, locations, organizations) in unstructured text (Nadeau and Sekine, 2007), is an essential prerequisite for many NLP applications. Being one of the most fundamental and classic sequence labeling tasks in NLP, there have been extensive research from traditional statistical models like Hidden Markov Models (Zhou and Su, 2002) and Conditional Random Fields (Lafferty et al., 2001a), to neural network based models such as LSTM-CRF (Lample et al., 2016a) and BLSTM-CNN-CRF (Ma and Hovy, 2016), and to recent pre-training and fine-tuning methods like ELMO (Peters et al., 2018a), Flair (Akbik et al., 2018) and BERT (Devlin et al., 2019). However, most of those models still heavily rely on *abundant annotated data* to yield the state-of-the-art results (Lin et al., 2020), making them hard to be applied into new domains (e.g., social media, medical context or low-resourced languages) that lack labeled data.

Different kinds of data augmentation approaches have been designed to alleviate the dependency on labeled data for many NLP tasks, and can be categorized into two broad classes: (1) adversarial attacks at token-levels such as word substitutions (Kobayashi, 2018; Wei and Zou, 2019) or adding noise (Lakshmi Narayan et al., 2019), (2) paraphrasing at sentence-levels such as back translations (Xie et al., 2019) or submodular optimized models (Kumar et al., 2019). The former has already been used for NER but struggles to create diverse augmented samples with very few word replacements. Despite being widely utilized in many NLP tasks like text classification, the latter often fails to maintain the labels at the token-level in those paraphrased sentences, thus making it difficult to be applied to NER.

We focus on another type of data augmentations called mixup (Zhang et al., 2018), which was originally proposed in computer vision and performed linear interpolations between randomly sampled image pairs to create virtual training data. Miao et al. (2020); Chen et al. (2020b) adapted the idea to textual domains and have applied it to the preliminary task of text classification. However, unlike classifications where each sentence only has one label, sequence labeling tasks such as NER usually involve multiple interrelated labels in a single sentence. As we found in empirical experiments, it is challenging to directly apply such mixup technique to sequence labeling, and improper interpolations may mislead the model. For instance, *random sam-*

*Equal contribution.

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 1241–1251,
November 16–20, 2020.

pling in mixup may inject too much noise by interpolating data points far away from each other, hence making it fail on sequence labeling.

To fill this gap, we propose a novel method called **L**ocal **A**dditivity based **D**ata **A**ugmentation **(LADA)**, in which we constrain the samples to mixup to be *close* to each other. Our method has two variations: **Intra-LADA** and **Inter-LADA**. Intra-LADA interpolates each token's hidden representation with other tokens from the same sentence, which could increase the robustness towards word orderings. Inter-LADA interpolates each token's hidden representation in a sentence with each token from other sentences sampled from a weighted combination of k-nearest neighbors sampling and random sampling, the weight of which controls the delicate trade-off between noise and regularization. To further enhance the performance of learning with limited labeled data, we extend LADA to the semi-supervised setting, i.e., **Semi-LADA**, by designing a novel consistency loss between unlabeled data and its local augmentations. We conduct experiments on two NER datasets to demonstrate the effectiveness of our LADA based models over state-of-the-art baselines.

2 Background

Zhang et al. (2018) proposed a data augmentation technique called *mixup*, which trained an image classifier on linear interpolations of randomly sampled image data. Given a pair of data points $(\mathbf{x}, \mathbf{y})$ and $(\mathbf{x}', \mathbf{y}')$, where $\mathbf{x}$ denotes an image in raw pixel space, and $\mathbf{y}$ is the label in a one-hot representation, *mixup* creates a new sample by interpolating images and their corresponding labels:

$$\tilde{\mathbf{x}} = \lambda \mathbf{x} + (1-\lambda)\mathbf{x}',$$
$$\tilde{\mathbf{y}} = \lambda \mathbf{y} + (1-\lambda)\mathbf{y}',$$

where λ is drawn from a Beta distribution. *mixup* trains the neural network for image classification by minimizing the loss on the virtual examples. In experiments, the pairs of images data points $(\mathbf{x}, \mathbf{y})$ and $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ are *randomly* sampled. By assuming all the images are mapped to a low dimension manifold through a neural network, linearly interpolating them creates a virtual vicinity distribution around the original data space, thus improving the generalization performance of the classifier trained on the interpolated samples.

Prior work like Snippext (Miao et al., 2020), MixText (Chen et al., 2020b) and AdvAug (Cheng et al., 2020) generalized the idea to the textual domain by proposing to interpolate in output space (Miao et al., 2020), embedding space (Cheng et al., 2020), or general hidden space (Chen et al., 2020b) of textual data and applied the technique to NLP tasks such as text classifications and machine translations and achieved significant improvements.

3 Method

Based on the above interpolation based data augmentation techniques, in Section 3.1, we introduced a **L**ocal **A**dditivity based **D**ata **A**ugmentation **(LADA)** for sequence labeling, where creating augmented samples is much more challenging. We continue to describe how to utilize unlabeled data with LADA for semi-supervised NER in Section 3.4.

3.1 LADA

For a given sentence with n tokens $\mathbf{x} = \{x_1, ..., x_n\}$, denote the corresponding sequence label as $\mathbf{y} = \{y_1, ..., y_n\}$. In this paper, we use NER as the working example to introduce our model, in which the labels are the entities types. We randomly sample a pair of sentences from the corpus, $(\mathbf{x}, \mathbf{y})$ and $(\mathbf{x}', \mathbf{y}')$, and then compute the interpolations in the hidden space using a L-layer encoder $\mathbf{F}(.;\theta)$. The hidden representations of $\mathbf{x}$ and $\mathbf{x}'$ up to the m-th layer are given by:

$$\mathbf{h}^l = \mathbf{F}^l(\mathbf{h}^{l-1};\theta), l \in [1, m],$$
$$\mathbf{h}'^l = \mathbf{F}^l(\mathbf{h}'^{l-1};\theta), l \in [1, m],$$

Here $\mathbf{h}^l = \{h_1, ..., h_n\}$ refer to the hidden representations at the l-th layer and is the concatenation of token representations at all positions. We use $\mathbf{h}^0, \mathbf{h}'^0$ to denote the word embedding of $\mathbf{x}$ and $\mathbf{x}'$ respectively. At the m-th layer, the hidden representations for each token in $\mathbf{x}$ are linearly interpolated with each token in $\mathbf{x}'$ by a ratio λ:

$$\tilde{\mathbf{h}}^m = \lambda \mathbf{h}^m + (1-\lambda)\mathbf{h}'^m,$$

where the mixing parameter λ is sampled from a Beta distribution, i.e., $\lambda \sim \text{Beta}(\alpha, \alpha)$. Then $\tilde{\mathbf{h}}^m$ is fed to the upper layers:

$$\tilde{\mathbf{h}}^l = \mathbf{F}^l(\tilde{\mathbf{h}}^{l-1};\theta), l \in [m+1, L].$$

$\tilde{\mathbf{h}}^L$ can be treated as the hidden representations of a *virtual sample* $\tilde{\mathbf{x}}$, i.e., $\tilde{\mathbf{h}}^L = \mathbf{F}(\tilde{\mathbf{x}};\theta)$.

In the meanwhile, their corresponding labels are linearly added with the same ratio:

$$\tilde{y}_i = \lambda y_i + (1-\lambda) y'_i$$
$$\tilde{\mathbf{y}} = \{\tilde{y}_1, ..., \tilde{y}_n\}.$$

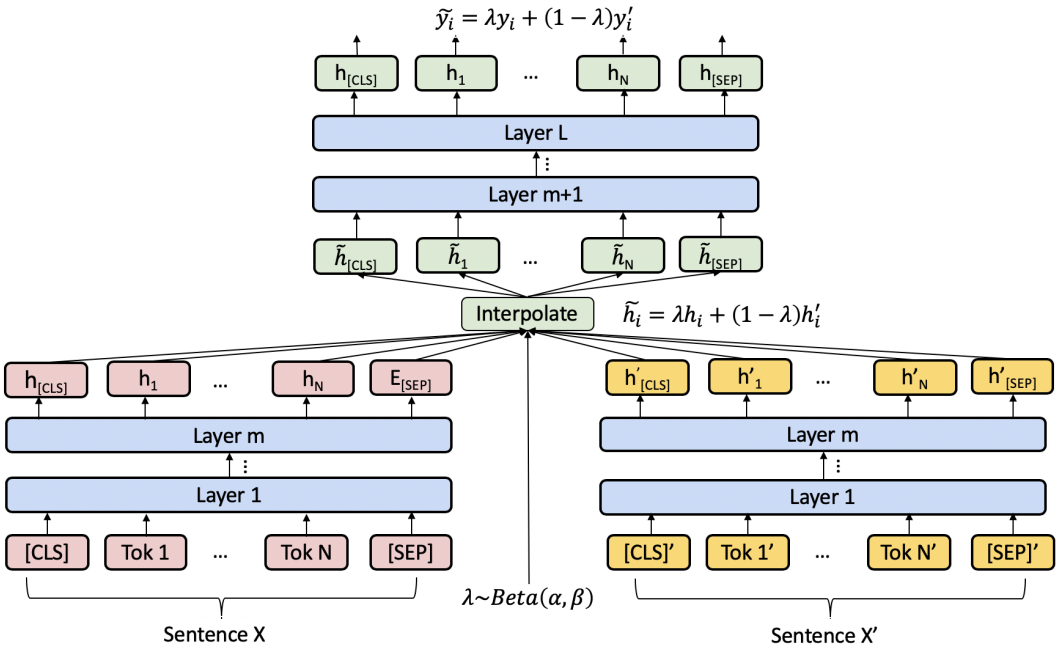

Figure 1: Overall Architecture of LADA. LADA takes in two sentences, linearly interpolates their hidden states h_i and h'_i at layer m with weight λ into $\tilde{h}_i$, and then continues forward passing to get encoded representations $\tilde{h}_i$, which are utilized in downstream tasks where the labels in each task are also mixed with weight λ.

The hidden representations $\tilde{\mathbf{h}}^L$ are then fed into a classifier $p(:, \phi)$ and the loss over all positions is minimized to train the model:

$$L = \mathop{\mathbb{E}}_{\mathbf{x}' \sim P_{\text{mix}}(\mathbf{x}'|\mathbf{x})} [\sum_{i=1}^{n} \text{KL}(\tilde{y}_i; p(\tilde{h}_i^L; \phi))]. \quad (1)$$

Here $P_{\text{mix}}(\mathbf{x}'|\mathbf{x})$ defines the probability of sampling $(\mathbf{x}', \mathbf{y}')$ to mix with $(\mathbf{x}, \mathbf{y})$. The overall diagram is shown in Figure 1.

Let $\mathbf{S} = \{(\mathbf{x}, \mathbf{y})\}$ be the corpus of data samples, then according to Chen et al. (2020b),

$$P_{\text{mix}}(\mathbf{x}'|\mathbf{x}) = \frac{1}{|\mathbf{S}|}, \qquad (\mathbf{x}', \mathbf{y}') \in \mathbf{S}. \quad (2)$$

Note that $P_{\text{mix}}(\mathbf{x}'|\mathbf{x})$ is a uniform distribution that is independent of $\mathbf{x}$. Even though $\mathbf{x}'$ can be far away from $\mathbf{x}$ in the Euclidean space, they are mapped into a low-dimension manifold through a neural network. Interpolating them in the hidden space regularizes the model to perform linearly in the low-dimensional manifold, hence greatly improves tasks such as classification.

However, we found empirically in experiments that the above *random sampling* strategy failed on sequence labeling like NER, leading to worse modeling results than purely supervised learning. Intuitively, sequence labeling is more complicated than sentence classification as it requires learning much more fine-grained information. Labeling a token depends on not only the token itself but also the *context*. We hypothesize that mixing the sequence $\mathbf{x}$ with $\mathbf{x}'$ changes the context for all tokens and injects too much noise, hence making learning the labels for the tokens challenging. In other words, the relative distance between $\mathbf{x}$ and $\mathbf{x}'$ in the manifold mapped by neural networks is further in sequence labeling than sentence classification (demonstrated in Figure 2), which is intuitively understandable as every data point in sentence classification is the pooling over all the tokens in one sentence while every token is a single data point in sequence labeling. Randomly mixing data points far away from each other introduces more noise for sequence labeling. To overcome this problem, we introduce a *local* additivity based data augmentation approach with two variations, in which we constrain $\mathbf{x}'$ to be close to $\mathbf{x}$:

3.2 Intra-LADA

As stated above, mixing two sequences not only changes the local token representations but also affects the context required to label tokens. To reduce the noises from unrelated sentences, the most direct way is to construct $\mathbf{x}'$ using the same tokens from $\mathbf{x}$ but changing the orders and perform interpolations between them.

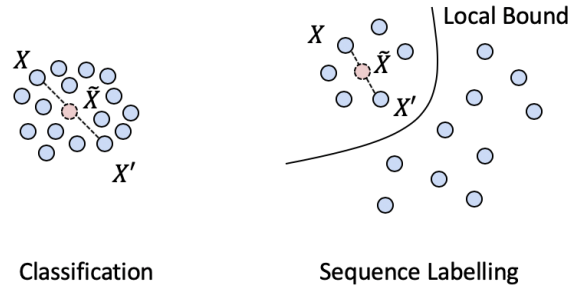

Figure 2: Data manifold for sentence classification and sequence labeling. The dimension of data manifold for sequence labeling is higher than sentence classification, hence the distance between data samples is larger. We constraint $\mathbf{x}'$ to be close to $\mathbf{x}$ in creating interpolated data in LADA.

Let $\mathbf{Q} = \text{Permutations}((\mathbf{x}, \mathbf{y}))$ be the set including all possible permutations of $\mathbf{x}$, then

$$P_{\text{Intra}}(\mathbf{x}'|\mathbf{x}) = \frac{1}{n!}, \qquad (\mathbf{x}', \mathbf{y}') \in \mathbf{Q}. \quad (3)$$

In this case, each token x_i in $\mathbf{x}$ is actually interpolated with another token x_j in $\mathbf{x}$, while the context is unaltered. By sampling from P_{Intra}, we are essentially turning sequence level interpolation to token level interpolation, thus greatly reducing the complexity of the problem. From another perspective, Intra-LADA generates augmentations with different sentence structures using the same word set, which could potentially increase the model's robustness towards word orderings.

Intra-LADA restraints the context from changing, which could be limited in generating diverse augmented data. To overcome that, we propose Inter-LADA, where we sample a different sentence from the training set to perform interpolations.

3.3 Inter-LADA

Instead of interpolating within one sentence, Intra-LADA samples a different sentence $\mathbf{x}'$ from the training set to interpolate with $\mathbf{x}$. To achieve a trade-off between noise and regularization, we sample $\mathbf{x}'$ through a weighted combination of two strategies: k-nearest neighbors (kNNs) sampling and random sampling:

$$P_{\text{Inter}}(\mathbf{x}'|\mathbf{x}) = \begin{cases} \frac{\mu}{k}, & \mathbf{x}' \in \text{Neighbor}_k(\mathbf{x}), \\ \frac{1-\mu}{|S|}, & (\mathbf{x}', \mathbf{y}') \in \mathbf{S}, \end{cases} \quad (4)$$

where μ is the weight of combining two distributions. To get the kNNs, we use sentence-BERT (Reimers and Gurevych, 2019) to map each sentence $\mathbf{x}$ into a hidden space, then collect each sentence's kNNs using l^2 distance. For each sentence $\mathbf{x}$, we sample $\mathbf{x}'$ to mix up from the kNNs with probability μ and the whole training corpus with a probability $1 - \mu$. When $\mathbf{x}'$ is sampled from the whole training corpus, it may be unrelated to $\mathbf{x}$, introducing large noise but also strong regularization on the model. When $\mathbf{x}'$ is sampled from the kNNs, $\mathbf{x}'$ shares similar, albeit different, context with $\mathbf{x}$, thus achieving good signal to noise ratio. By treating μ as a hyper-parameter, we can control the delicate trade-off between noise and diversity in regularizing the model.

To examine why sampling sentences from kNNs decreases the noise and provides meaningful signals to training, we analyze an example with its kNNs in Table 1: (1) As it shows, kNNs may contain the same entity words as the original sentence, but in different contexts. The entity types in the neighbor sentences are also changed corresponding to contexts. For example, entity *Israel* in the third neighbor becomes an organization when surrounded by *Radio* while it is a location in the original sentence. (2) Contexts from neighbor sentences can help detect the entities of the same type in a given sentence. For example, *Lebanon* in the second neighbor shares the same type as *Israel* in the original sentence. *Lebanon* can resort to the context of the original sentence to detect its entity type. (3) Neighbor sentences may contain the same words but in different forms. For example, the *Israeli* in the first neighbor sentence is a different form of *Israel*, which is miscellaneous while *Israel* is a location in the example sentence. Interpolation with such an example can improve models' ability to recognize words of different forms and their corresponding types.

In summary, Inter-LADA can improve both entity learning and context learning by interpolating more diverse data. Note that although we use NER as a working example , LADA can be applied to any sequence labeling models.

3.4 Semi-supervised LADA

To further improve the performance of learning with less labeled data, we propose a novel LADA-based approach specifically for unlabeled data. Instead of looking for nearest neighbors, we use back-translation techniques to generate paraphrases of an unlabeled sentence $\mathbf{x}_u$ in constructing $\mathbf{x}'_u$. The paraphrase $\mathbf{x}'_u$, generated via translating $\mathbf{x}_u$ to an intermediate language and then translating it back, describes the same content as $\mathbf{x}_u$ and should be close to $\mathbf{x}_u$ semantically. However, there is no

Sentence	Israel plays down fears of war with Syria.
Neighbours	Fears of an Israeli operation causes the redistribution of Syrian troops locations in Lebanon .
	Parliament Speaker Berri: Israel is preparing for war against Syria and Lebanon .
	Itamar Rabinovich , who as Israel's ambassador to Washington conducted unfruitful negotiations with Syria , told Israel Radio looked like Damascus wanted to talk rather than fight .

Table 1: kNNs of an example sentence. Entities in sentences are colored. Green means locations , red means persons , blue means organizations and yellow means miscellaneous.

guarantee that the same entity would appear in the same position in $\mathbf{x}_u$ and $\mathbf{x}'_u$. In fact, the number of tokens in $\mathbf{x}_u$ and $\mathbf{x}'_u$ may not even be the same. For instance, for the sentence "*Rare **Hendrix** song draft sells for almost $17,000*" and its paraphrased sentence "*A rare **Hendrix** song design is selling for just under $17,000*", although some words are different, the entity *Hendrix* keeps unchanged, and there are no extra entities added. That is, both contain one and only one entity (*Hendrix*) of the same type (*Person*). Nevertheless, we empirically found that most paraphrases contain the same number of entities (for any specific type) as the original sentence. Inspired by the observation, we propose a new consistency loss to leverage unlabeled data: $\mathbf{x}_u$ and $\mathbf{x}'_u$ should have the same number of entities for any given entity type.

Specifically, for an unlabeled sentence $\mathbf{x}_u$ and its paraphrase $\mathbf{x}'_u$, we first guess their token labels with the current model:

$$\mathbf{y}_u = p(\mathbf{F}(\mathbf{x}_u; \theta); \phi).$$

To avoid predictions being too uniform at the early stage, we sharpen every token prediction $y_{u,i} \in \mathbf{y}_u$ with a temperature T:

$$\hat{y}_{u,i} = \frac{(y_{u,i})^{\frac{1}{T}}}{\left\|(y_{u,i})^{\frac{1}{T}}\right\|_1},$$

where $||.||_1$ denotes the $l1$-norm. We then add the prediction $\hat{y}_{u,i}$ over all tokens in the sentence to denote its total number of entities for each type:

$$\hat{y}_{u,\,\text{num}} = \sum_{i=1}^{n} \hat{y}_{u,i}.$$

Note that $\hat{y}_{u,\text{num}}$ is the guessed label vector with C-dimensions, where C is the total number of entity types. The i-th element in the $\hat{y}_{u,\text{num}}$ denotes the total number i-type entity in the sentence.

Dataset	CoNLL	GermEval
Train	14,987	24,000
Dev	3,466	2,200
Test	3,684	5,100
Entity Types	4	12
Max Sent Length	142	84

Table 2: Data statistics and our data split following Benikova et al. (2014).

During training, we use the same procedure to get the number of entities for original and each paraphrase sentence (without sharpening). Assume there are K paraphrases, denote the entity number vector for the k-the paraphrase as $\hat{y}'^{k}_{u,\text{num}}$. The consistency objective for unlabeled sentence $\mathbf{x}$ and its paraphrases is:

$$L_u = ||\hat{y}_{u,\text{num}} - \hat{y}'^{k}_{u,\text{num}}||^2. \quad (5)$$

Here we treat $\hat{y}_{u,\text{num}}$ as fixed and back-propagate only through $\hat{y}'_{u,\text{num}}$ to train the model.

Taking into account the loss objectives for both labeled and unlabeled data (Equation 1 and Equation 5), our **Semi-LADA** training objective is:

$$L_{semi} = L + \gamma L_u$$

where γ controls the trade-off between the supervised loss term and the unsupervised loss term.

4 Experiments

4.1 Datasets and Pre-processing

We performed experiments on two datasets in different languages: **CoNLL** 2003 (Tjong Kim Sang and De Meulder, 2003) in English and **GermEval** 2014 (Benikova et al., 2014) in German. The data statistics are shown in Table 2. We used the BIO labeling scheme and reported the F1 score. In order to make LADA possible in recent transformer-based models like BERT, we assigned labels to

Model	Unlabeled data	CoNLL			GermEval		
		5%	10%	30%	5%	10%	30%
Flair (Akbik et al., 2019)	no	79.32	86.31	89.96	66.54	67.92	74.11
Flair + Intra-LADA†	no	-	-	-	-	-	-
Flair + Inter-LADA†	no	80.84	86.33	**90.61**	67.40	70.02	74.63
BERT (Devlin et al., 2019)	no	83.28	86.85	89.28	79.64	80.92	82.87
BERT + Intra-LADA†	no	83.52	87.54	89.31	79.93	81.10	82.92
BERT + Inter-LADA†	no	84.60	87.81	89.68	80.13	**81.28**	83.63
BERT + Intra&Inter-LADA†	no	**84.85**	**87.85**	89.87	**80.17**	81.23	**83.65**
VSL-GG-Hier (Chen et al., 2018)	yes	83.38	84.71	85.52	-	-	-
MT + Noise (Lakshmi Narayan et al., 2019)	yes	82.60	83.47	84.88	-	-	-
BERT + Semi-Intra-LADA†	yes	**87.15**	88.70	89.69	80.95	81.52	83.46
BERT + Semi-Inter-LADA†	yes	86.51	88.53	90.00	**81.20**	81.70	83.53
BERT + Semi-Intra&Inter-LADA†	yes	86.33	**88.78**	**90.25**	81.07	**81.77**	**83.63**

Table 3: The F1 scores on CoNLL 2003 and GermEval 2014 training with varying amounts of the labeled training data (5%, 10%, and 30% of the original training set). There were 10,000 unlabeled data for each dataset which was randomly sampled from the original training set. All the results were averaged over 5 runs. † denotes our methods.

special tokens [SEP], [CLS], and [PAD]. Since BERT tokenized a token into one or multiple sub-tokens, we not only assigned labels to the first sub-token but also to the remaining sub-tokens following the rules: (1) O word: Oxx→OOO, (2) I word: Ixx→III,(3) B word: Bxx→BII, as such kind of assignment will not harm the performance (ablation study was conducted in Section 4.4). During the evaluation, we ignored special tokens and non-first sub-tokens for fair comparisons.

In the fully supervised setting, we followed the standard data splits shown in Table 2. In the semi-supervised setting, we sampled 10,000 sentences in the training set as the unlabeled training data. We adopted FairSeq[1] to implement the back translation. For CoNLL dataset, we utilized German as the intermediate language and English as the intermediate language for GermEval.

4.2 Baselines & Model Settings

Our LADA can be applied to any models in standard sequence labeling frameworks. In this work, we applied LADA to two state-of-the-art pre-trained models to show the effectiveness:

- **Flair** (Akbik et al., 2019): We used the pre-trained Flair embeddings[2], and a multi-layer BiLSTM-CRF (Ma and Hovy, 2016) as the encoder to detect the entities.
- **BERT** (Devlin et al., 2019): We loaded the BERT-base-multilingual-cased[3] as the encoder and a linear layer to predict token labels.

To demonstrate whether our Semi-LADA works with unlabeled data, we compared it with two recent state-of-the-art semi-supervised NER models:

- **VSL-GG-Hier** (Chen et al., 2018) introduced a hierarchical latent variables models into semi-supervised NER learning.
- **MT + Noise** (Lakshmi Narayan et al., 2019) explored different noise strategies including word-dropout, synonym-replace, Gaussian noise and network-dropout in a mean-teacher framework.

We also compared our models with another two recent state-of-the-art NER models trained on the whole training set:

- **CVT** (Clark et al., 2018) performed multi-task learning and made use of 1 Billion Word Language Model Benchmark as the source of unlabeled data.
- **BERT-MRC** (Li et al., 2020) formulated the NER as a machine reading comprehension task instead of a sequence labeling problem.

For **Intra-LADA**, as it broke the sentence structures, it cannot be applied to Flair that was based on LSTM-CRF. Thus we only combined it with BERT and only used the labeled data. The mix layer set was $\{12\}$. For **Inter-LADA**, we applied it to Flair and BERT trained with only the labeled data. The mix layer set was $\{8,9,10\}$, k in kNNs was 3, and 0.5 was a good start point for tuning μ. **Semi-LADA** utilized unlabeled data as well. The model was built on BERT. The weight γ to balance the supervised loss and unsupervised loss was 1.

[1] https://github.com/pytorch/fairseq
[2] https://github.com/flairNLP/flair
[3] https://github.com/huggingface/transformers

Model	Setting	CoNLL	GermEval
Flair (Akbik et al., 2019)	Token Classification	92.03	76.92
Flair + Intra-LADA ‡	Token Classification	-	-
Flair + Inter-LADA ‡	Token Classification	**92.12**	**78.45**
BERT (Devlin et al., 2019)	Token Classification	91.19	86.12
BERT + Intra-LADA ‡	Token Classification	91.22	86.16
BERT + Inter-LADA ‡	Token Classification	**91.83**	**86.45**
CVT (Clark et al., 2018)	Multi-task Learning	92.60	-
BERT-MRC (Li et al., 2020)	Reading Comprehension	93.04	-

Table 4: The F1 score on CoNLL 2003 and GermEval 2014 training with all the labeled training data. ‡ means incorporating our LADA data augmentation techniques into pre-trained models.

4.3 Main Results

We evaluated the baselines and our methods using F1-scores on the test set.

Utilizing Limited Labeled Data We varied the number of labeled data (made use of 5%, 10%, 30% of labeled sentences in each dataset, which were 700, 1400, 4200 in CoNLL and 1200, 2400, 7200 in GermEval) and the results were shown in Table 3. Compared to purely Flair and BERT, applying *Intra-LADA* and *Inter-LADA* consistently boosted performances significantly, indicating the effectiveness of creating augmented training data through local linear interpolations. When unlabeled data was introduced, VSL-GG-Hier and MT + Noise performed slightly better than Flair and BERT with 5% labeled data in CoNLL, but pre-trained models (Flair, BERT) still got higher F1 scores when there were more labeled data. Both kinds of *BERT + Semi-LADA* significantly boosted the F1 scores on CoNLL and GermEval compared to baselines, as Semi-LADA not only utilized LADA on labeled data to avoid overfitting but also combined back translation based data augmentations on unlabeled data for consistent training, which made full use of both labeled data and unlabeled data.

Utilizing All the Labeled data Table 4 summarized the experimental results on the full training sets (14,987 on CoNLL 2003 and 24,000 on GermEval 2014). Compared to pre-trained Flair and BERT[4], there were still significant performance gains from utilizing our LADA, which indicated that our proposed data augmentation methods work well even with a large amount of labeled training data (full datasets). We also showed two state-of-the-art NER models' results with different settings, they had better performance mainly due to the multi-task learning with more unlabeled data (CVT) or formulating the NER as reading comprehension problems (BERT + MRC). Note that our LADA was orthogonal to these two models.

Loss on the Development Set To illustrate that our LADA could also help the overfitting problem, we plotted the loss on the development set of BERT, *BERT + Inter-LADA* and *BERT + Semi-Inter-LADA* on CoNLL and GermEval training with 5% labeled data in Figure 3. After applying LADA, the loss curve was more stable with training epoch increased, while the loss curve of BERT started increasing after about 10 epochs, indicating that the model might overfit the training data. Such property made LADA a suitable method, especially for semi-supervised learning.

Combining Intra&Inter-LADA We further combined Intra-LADA and Inter-LADA with a ratio π, i.e. data point would be augmented through Intra-LADA with a probability π and Inter-LADA with a probability $1 - \pi$. In practice, we set the probability 0.3, and kept the settings for each kind of LADA the same. The results are shown in Table 3. Through combining two variations, *BERT + Intra&Inter-LADA* further boosted model performance on both datasets, with an increase of 0.25, 0.04 and 0.19 on CoNLL over *BERT + Inter-LADA* trained with 5%, 10% and 30% labeled data. We obtained consistent improvement in semi-supervised settings: *BERT + Semi-Intra&Inter-LADA* improved over *BERT*

[4] Note that for the discrepancy between our BERT results and results published in the BERT paper, it has been discussed in the official repo https://github.com/google-research/bert/issues/223, where the best performance one can replicate on CoNLL was around 91.3 based on the given scripts. For our experiments, we followed the provided scripts, and kept model settings identical as baselines for fair comparison.

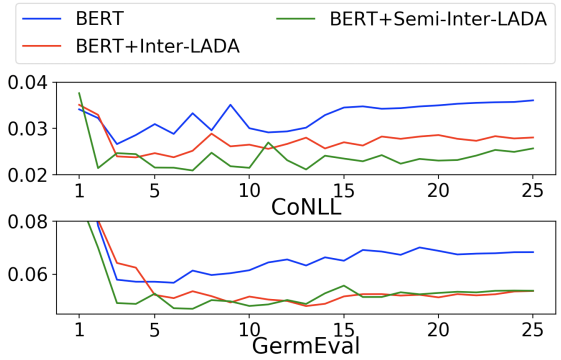

Figure 3: Loss (Y axis) on development set, trained with 5% labeled data, over different epochs (X axis).

+ *Semi-Inter-LADA* trained with 5%, 10% and 30% labeled data on GermEval by +0.05, +0.07 and +0.10. This showed that our Intra-LADA and Inter-LADA can be easily combined by future work to create diverse augmented data to help sequence labeling tasks.

4.4 Ablation Study

Different Sub-token Labeling Strategies To prove that our pre-processing of labeling sub-tokens for training was reasonable, we compared BERT training with different sub-token labeling strategies in Table 5."**None**" strategy was used in original BERT-Tagger where sub-tokens are ignored during learning. "**Real**" strategy was used in our Inter-LADA where O words' sub-tokens were assigned O (Oxx→OOO), I and B words' sub-tokens were assigned I (Ixx→III, Bxx→BII). "**Repeat**" referred to assigning the original label to each sub-token (Oxx→OOO, Ixx→III, Bxx→BBB). "O" means we assigned O to each sub-token (Oxx→OOO, Ixx→IOO, Bxx→BOO). "Real" strategy received comparable performances with original BERT models while the other two strategies decreased F1 scores, indicating our strategy mitigated the sub-token labeling issue.

Influence of μ in Inter-LADA We varied the μ in *BERT + Inter-LADA* from 0 to 1 to validate that combining kNNs sampling and random sampling in Inter-LADA could achieve the best performance, and the results were plotted in Figure 4. Note that when $\mu = 0$, Inter-LADA only did random sampling and it barely improved over BERT largely due to too much noise from interpolations between unrelated sentences. And when $\mu = 1$, Inter-LADA only did kNNs sampling, and it could get a better F1 score over BERT because of providing meaningful signals to training. BERT + Inter-LADA got the best F1 score with $\mu = 0.7$ on CoNLL and $\mu = 0.5$ on GermEval, which indicated the trade-off between noise and diversity (kNNs sampling with lower noise and random sampling with higher diversity) was necessary for Inter-LADA.

Tag Strategy	CoNLL	GermEval
None	83.28	79.64
Real	84.15	79.59
Repeat	82.67	78.27
O	83.13	78.48

Table 5: F1 scores of BERT on test set with different strategy to tag sub-tokens trained with 5% labeled data.

5 Related Work

5.1 Named Entity Recognition

Conditional random fields (CRFs) (Lafferty et al., 2001b; Sutton et al., 2004) have been widely used for NER, until recently they have been outperformed by neural networks. Hammerton (2003) and Collobert et al. (2011) are among the first several studies to model sequence labeling using neural networks. Specifically Hammerton (2003) encoded the input sequence using a unidirectional LSTM (Hochreiter and Schmidhuber, 1997) while (Collobert et al., 2011) instead used a CNN with character level embedding to encode sentences. Ma and Hovy (2016); Lample et al. (2016b) proposed LSTM-CRFs to combine neural networks with CRFs that aim to leverage both the representation learning capabilities of neural network and structured loss from CRFs. Instead of modeling NER as a sequence modeling problem, Li et al. (2020) converted NER into a reading comprehension task with an input sentence and a query sentence based on the entity types and achieved competitive performance.

5.2 Semi-supervised Learning for NER

There has been extensive previous work (Altun et al., 2005; Søgaard, 2011; Mann and McCallum, 2010) that utilized semi-supervised learning for NER. For instance, (Zhang et al., 2017; Chen et al., 2018) applied variational autoencoders (VAEs) to semi-supervised sequence labeling; (Zhang et al., 2017) proposed to use discrete labeling sequence as latent variables while (Chen et al., 2018) used continuous latent variables in their models. Recently, contextual representations such as ELMO (Peters

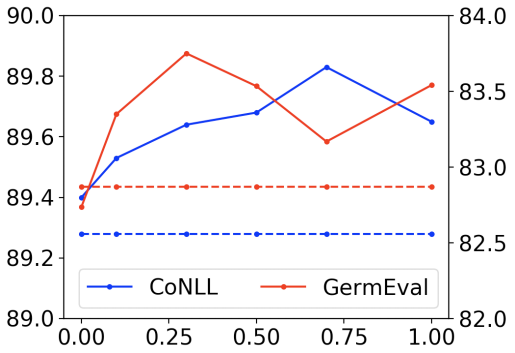

Figure 4: F1 score on test set training with 30% labeled data with different μ in BERT + Inter-LADA. The left Y axis is for CoNLL, and the right Y axis is for GermEval. Dashed lines are the F1 scores of BERT model.

et al., 2018b) and BERT (Devlin et al., 2019) trained on a large amount of unlabeled data have been applied to NER and achieved reasonable performances. Our work is related to research that introduces different data augmentation techniques for NER. For example, Lakshmi Narayan et al. (2019) applied noise injection and word dropout and obtained a performance boost, Bodapati et al. (2019) varied the capitalization of words to increase the robustness to capitalization errors, Liu et al. (2019) augmented traditional models with pretraining on external knowledge bases. In contrast, our work can be viewed as data augmentation in the continuous hidden space without external resources.

5.3 Mixup-based Data Augmentation

Mixup (Zhang et al., 2018) was originally proposed for image classification (Verma et al., 2018; Yun et al., 2019) as a data augmentation and regularization method , building on which Miao et al. (2020) proposed to interpolate sentences' encoded representations with augmented sentences by token-substitutions for text classification. Similarly, Chen et al. (2020a) designed a linguistically informed interpolation of hidden space and demonstrated significant performance increases on several text classification benchmarks. Cheng et al. (2020) performed interpolations at the embedding space in sequence-to-sequence learning for machine translations. Different from these previous studies, we sample sentences based on *local* additivity and utilize mixup for the task of sequence labeling.

6 Conclusion

This paper introduced a local additivity based data augmentation (LADA) methods for Named Entity Recognition (NER) with two different interpolation strategies. To utilize unlabeled data, we introduced a novel consistent training objective combined with LADA. Experiments have been conducted and proved our proposed methods' effectiveness through comparing with several state-of-the-art models on two NER benchmarks.

Acknowledgment

We would like to thank the anonymous reviewers for their helpful comments, and the members of Georgia Tech SALT group for their feedback. We acknowledge the support of NVIDIA Corporation with the donation of GPU used for this research.

References

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Yasemin Altun, David A. McAllester, and Mikhail Belkin. 2005. Margin semi-supervised learning for structured variables. In *Advances in Neural Information Processing Systems 18 [Neural Information Processing Systems, NIPS 2005, December 5-8, 2005, Vancouver, British Columbia, Canada]*, pages 33–40.

Darina Benikova, Chris Biemann, Max Kisselew, and Sebastian Pad. 2014. GermEval 2014 Named Entity Recognition Shared Task: Companion Paper. In *Proceedings of the KONVENS GermEval workshop*, pages 104–112, Hildesheim, Germany.

Sravan Bodapati, Hyokun Yun, and Yaser Al-Onaizan. 2019. Robustness to capitalization errors in named entity recognition. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 237–242, Hong Kong, China. Association for Computational Linguistics.

Jiaao Chen, Yuwei Wu, and Diyi Yang. 2020a. Semi-supervised models via data augmentationfor classifying interactive affective responses.

Jiaao Chen, Zichao Yang, and Diyi Yang. 2020b. Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification.

In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Seattle, Washington, USA. Association for Computational Linguistics.

Mingda Chen, Qingming Tang, Karen Livescu, and Kevin Gimpel. 2018. Variational sequential labelers for semi-supervised learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 215–226, Brussels, Belgium. Association for Computational Linguistics.

Yong Cheng, Lu Jiang, Wolfgang Macherey, and Jacob Eisenstein. 2020. Advaug: Robust data augmentation for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Seattle, Washington, USA. Association for Computational Linguistics.

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

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537.

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.

James Hammerton. 2003. Named entity recognition with long short-term memory. 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 172–175. ACL.

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

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457, New Orleans, Louisiana. Association for Computational Linguistics.

Ashutosh Kumar, Satwik Bhattamishra, Manik Bhandari, and Partha Talukdar. 2019. Submodular optimization-based diverse paraphrasing and its effectiveness in data augmentation. 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 3609–3619, Minneapolis, Minnesota. Association for Computational Linguistics.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001a. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML 01, page 282289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001b. 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. Morgan Kaufmann.

Pooja Lakshmi Narayan, Ajay Nagesh, and Mihai Surdeanu. 2019. Exploration of noise strategies in semi-supervised named entity classification. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 186–191, Minneapolis, Minnesota. Association for Computational Linguistics.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016a. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016b. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020. A unified mrc framework for named entity recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Seattle, Washington, USA. Association for Computational Linguistics.

Bill Yuchen Lin, Dong-Ho Lee, Ming Shen, Ryan Moreno, Xiao Huang, Prashant Shiralkar, and Xiang Ren. 2020. Triggerner: Learning with entity triggers as explanations for named entity recognition.

Angli Liu, Jingfei Du, and Veselin Stoyanov. 2019. Knowledge-augmented language model and its application to unsupervised named-entity recognition. 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 1142–1150, Minneapolis, Minnesota. Association for Computational Linguistics.

Xuezhe Ma and Eduard 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 (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

Gideon S. Mann and Andrew McCallum. 2010. Generalized expectation criteria for semi-supervised learning with weakly labeled data. *J. Mach. Learn. Res.*, 11:955–984.

Zhengjie Miao, Yuliang Li, Xiaolan Wang, and Wang-Chiew Tan. 2020. Snippext: Semi-supervised opinion mining with augmented data. In *Proceedings of The Web Conference 2020*, WWW 20, page 617628, New York, NY, USA. Association for Computing Machinery.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26. Publisher: John Benjamins Publishing Company.

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

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

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Anders Søgaard. 2011. Semi-supervised condensed nearest neighbor for part-of-speech tagging. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 48–52, Portland, Oregon, USA. Association for Computational Linguistics.

Charles A. Sutton, Khashayar Rohanimanesh, and Andrew McCallum. 2004. Dynamic conditional random fields: factorized probabilistic models for labeling and segmenting sequence data. In *Machine Learning, Proceedings of the Twenty-first International Conference (ICML 2004), Banff, Alberta, Canada, July 4-8, 2004*, volume 69 of *ACM International Conference Proceeding Series*. ACM.

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 at HLT-NAACL 2003 - Volume 4*, CONLL 03, page 142147, USA. Association for Computational Linguistics.

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, Aaron Courville, David Lopez-Paz, and Yoshua Bengio. 2018. Manifold mixup: Better representations by interpolating hidden states. *arXiv preprint arXiv:1806.05236*.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Qizhe Xie, Zihang Dai, Eduard H. Hovy, Minh-Thang Luong, and Quoc V. Le. 2019. Unsupervised data augmentation. *CoRR*, abs/1904.12848.

Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Seong Joon Oh, Youngjoon Yoo, and Junsuk Choe. 2019. Cutmix: Regularization strategy to train strong classifiers with localizable features. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*.

Xiao Zhang, Yong Jiang, Hao Peng, Kewei Tu, and Dan Goldwasser. 2017. Semi-supervised structured prediction with neural CRF autoencoder. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1701–1711, Copenhagen, Denmark. Association for Computational Linguistics.

GuoDong Zhou and Jian Su. 2002. Named entity recognition using an HMM-based chunk tagger. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 473–480, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.